Temporal models of streaming social media data

The University Of Sheffield.

Daniel Preoţiuc-Pietro
Department of Computer Science
The University of Sheffield

A thesis submitted for the degree of
Doctor of Philosophy

June 2014

Abstract

There are significant temporal dependencies between online behaviour and occurring real world activities. Particularly in text modelling, these are usually ignored or at best dealt with in overly simplistic ways such as assuming smooth variation with time. Social media is a new data source which present collective behaviour much more richly than traditional sources, such as newswire, with a finer time granularity, timely reflection of activities, multiple modalities and large volume. Analysing temporal patterns in this data is important in order to discover newly emerging topics, periodic occurrences and correlation or causality to real world indicators or human behaviour patterns. With these opportunities come many challenges, both engineering (i.e. data volume and processing) and algorithmic, namely the inconsistency and short length of the messages and the presence of large amounts of irrelevant messages to our goal. Equipped with a better understanding of the dynamics of the complex temporal dependencies, tasks such as classification can be augmented to provide temporally aware responses.

In this thesis we model the temporal dynamics of social media data. We first show that temporality is an important characteristic of this type of data. Further comparisons and correlation to real world indicators show that this data gives a timely reflection of real world events. Our goal is to use these variations to discover emerging or recurring user behaviours. We consider both the use of words and user behaviour in social media. With these goals in mind, we adapt existing and build novel machine learning techniques. These span a wide range of models: from Markov models to regularised regression models and from evolutionary spectral clustering which models smooth temporal variation to Gaussian Process regression which can identify more complex temporal patterns.

We introduce approaches which discover and predict words, topics or behaviours that change over time or occur with some regularity. These

are modeled for the first time in the NLP literature by using Gaussian Processes. We demonstrate that we can effectively pick out patterns, including periodicities, and achieve state-of-the-art forecasting results. We show that this performance gain transfers to improve tasks which do not take temporal information in account. Further analysed is how temporal variation in the text can be used to discover and track new content. We develop a model that exploits the variation in word co-occurrences for clustering over time. Different collection and processing tools, as well as several datasets of social media data have been developed and published as open-source software.

The thesis posits that temporal analysis of data, from social media in particular, provides us with insights into real-world dynamics. Incorporating this temporal information into other applications can benefit standard tasks in natural language processing and beyond.

Acknowledgements

I would first like to thank my supervisor, Trevor Cohn, for guiding me through this hard but rewarding process. His expertise, vision and drive had a major impact in my career as a researcher. He has provided me patience, understanding and great feedback when I needed it most.

The Department of Computer Science at the University of Sheffield and the Trendminer project offered me a great environment for research. I had the privilege to collaborate with great colleagues such as Vasileios Lampos, Sina Samangooei, Nikolaos Aletras, Andrea Varga, Kalina Bontcheva, and Dominic Rout.

I would also like to thank my thesis examiners, Lucia Specia and Nello Cristianini, for their expertise and attention to detail. Without their help, my thesis would be poorer. A special thanks is also for Noah Smith and Chris Dyer which have kindly invited me to visit Carnegie Mellon University during my studies. My experience there was invaluable and has heavily and positively impacted my research.

However, I would have never chosen this path without the time I spent at the University of Bucharest. Responsible for my career path was mostly Florentina Hristea who introduced me to AI and NLP in particular. The enthusiasm of my professors there, especially Marius Popescu, Liviu Dinu and Horia Georgescu, provided an inspiration for me.

On a different, personal scale, none of this was possible without the help of my parents, Valeriu Preoţiuc-Pietro and Ana-Romaniţa Preoţiuc-Pietro, and my girlfriend Ioana Pal. They have given me the support I needed while I was away for my studies.

Contents

Chapter 1

Introduction

In an Internet where user generated content is ubiquitous, systems that are aware of the temporal evolution of data are paramount for a better understanding of real world phenomena. Thus, in this thesis, we aim to develop machine learning models for streaming user generated social media data which incorporate the temporal component in order to discover topic emergence, recurrence or correlation to underlying events.

The Internet has become immense[1] and plays a large part in the life of most people. This is mainly due to its widespread and proliferation of devices that can access it. Many people have shifted much of their daily personal activities to the online medium like socialising, work or information gathering. This is particularly obvious by the recent burst in usage of social media and user-generated content. Reportedly, social media usage became the single most time-consuming activity on the web as measured within the U.S., overtaking search, e-mail or gaming.[2]

With social media, we have seen a shift in paradigms for posting online content. Before social media, and the so-called Web 2.0 in general, the content was static; there were a few hubs of information which were updated only by those who had the authority, curating the content in the process. Now, also with the aid of the technical advances, user-generated content has become prevalent online. Thus, every person using social media can be thought as being a 'journalist' [Murthy, 2011] on his daily life, exhibited by him sharing news, events, moods or images.

For many, these sources have become the primary source of information for time sensitive or local events. This is mostly because people trust their peers more for giving accurate more insights, comments and a complementary view compared to traditional editorial sources. The lack of curation tends to lead to swifter responses.

[1] http://archive.org/

[2] http://cn.nielsen.com/documents/Nielsen-Social-Media-Report_FINAL_090911.pdf

A notorious example illustrating the timely nature of social media (i.e. Twitter) updates was the 2011 Earthquake in Virginia, U.S. Allegedly, tweets about the earthquake were read by people living in nearby cities before the seismic waves reached their location.[3] Indeed, currently many people follow events [Johnson, 2009], like sports matches, foremost through the social media lens, as it provides very up-to-date and broad coverage which allows to get a better picture for the atmosphere and public perception of that event.

It is therefore not surprising that even more 'traditional' sources have embraced social media for time sensitive communications. For example, local police departments in the U.K. started in 2010 to offer emergency information and updates about public safety, disruptions or disorder through social media, recognising the importance for disseminating time sensitive information this way.

However, all these benefits come with some obvious downsides. Lack of content curation can lead to content that is unreliable or misleading. Malicious users or spammers might create content of hard to asses authenticity. From a practical perspective, the volume of the data is vast, making storage and processing challenging. Due to the breadth of the content, most will be probably irrelevant for any task while having badly structured information, non-literal text and lack of context.

Social media websites are built around users sharing a piece of content within the underlying social network. This piece of content can be either short text (e.g. Twitter), photos (e.g. Instagram, Pinterest), location (e.g. Foursquare), professional information (e.g. LinkedIn), videos (e.g. YouTube, Vimeo), business reviews (e.g. Yelp) or general personal information (e.g. Facebook, Google+). There are multiple peculiarities that make social media exciting for research such as the presence of the underlying social network structure, the multiple modalities of the data, the spatial annotations and its richness and diversity.

In this thesis we will focus on using the temporal information that enhances social media data. We argue that, conditioning on time, the data has different proprieties that can be harnessed for practical objectives. We discover that a part of this data is informative and reflects real world activities and trends. Before social media, textual and behavioural temporal data was harder to gather and was limited to a specific set of sources. By using social media data we can analyse text at a much finer granularity (cf. to news), of a broader set of topics (i.e. related to daily life, TV shows) and in particular, identify specific effects such as periodicities.

[3] `http://www.huffingtonpost.com/2011/09/01/twitter-ad-earthquake-video_n_945480.html`

We consider two broad scenarios for our models. Foremost, a supervised forecasting setting that aims to forecast either a variable that is internal to the social media system (e.g. future word frequencies) or an external real world variable that we expect to be correlated to features derived from social media (e.g. political party voting intention). We use historical time series to forecast future values using social media features that are independent of external resources and generalisable across use scenarios (e.g. language). Secondly, we consider an unsupervised scenario that explores time variations in social media text usage and groups words into clusters which can be linked to real world events that have initially caused these variations. Temporal smoothness constraints imposed on the clusters between consecutive time steps allows interpretation of topic drift.

The main challenges we have to deal with when using social media data are the volume and the non-standard form of the data. Also, social media is very diverse and used for a wide range of activities, from spam and self-promotion to chatter and news reporting. Usually a large amount of data is irrelevant to our goals and must be treated as input noise.

1.1 Aims

- The first aim of the thesis is to show that social media data is time dependent. We study data over time from two social media platforms: Twitter, which is used to share short texts, and Foursquare, which is used to share the location of the author with additional meta-data.

  The outcome is a solid motivation that, conditioning on time, a better understanding can be achieved of what happened or will likely happen at that specific moment. In addition, a justification for the importance of modelling complex temporal patterns, specifically periodicities, is presented. Other outcomes are models that correlate social media data with trends in order to forecast real world indicators. These explicitly incorporate the noise that is present in social media in the form of non-informative words and authors. Correlation with real world trends gives an indication on the time dependence of the data.

- The main aim of the thesis is to prove that modelling the temporal information of social media text is beneficial for gaining a better understanding of real world effects. Simple temporal patterns, such a smoothness, as well as complex periodic patters are studied.

The outcomes are models which can automatically identify temporal patterns in data from both of our sources, namely words and locations. Given the data conditioned on time, we aim to infer the temporal patterns from that data in order to forecast future values in a supervised setting. The forecasting methods go beyond state-of-the-art and use Gaussian Process regression in order to categorise the temporal patterns and forecast values.

Another outcome is a method that uses the changes in the word co-occurrence distributions to perform event detection and tracking over time. The presented method can identify and label very specific events and general topics, whilst also scaling to the large volume of data arising from social media. This is performed by using a spectral clustering algorithm which imposes clusters in consecutive time windows to be similar, imposing a temporal smoothness constraint. Temporal smoothness is also incorporated in predicting real world indicators.

- Another important aim is to show the practicality of modelling time by incorporating them downstream into applications, improving their accuracy.

  The outcomes are classification and summarisation applications which use the temporal information in order to improve accuracy and aid interpretation.

- The techniques developed in this thesis are aimed to be reproducible and portable to new use cases. We aim for a limited dependence on external information and specific features of any language.

  A peripheral outcome of the thesis is the development of a series of open-source tools for collecting and processing OSN data.

1.2 Thesis structure

The first two chapters of the thesis examine Online Social Networks (OSN) as the source of our analysis. Chapter 2 introduces OSNs and presents an overview of their properties. It also introduces the datasets we will use in the rest of the thesis and the methods for collecting the data. These are available as open-source software packages. Chapter 3 describes the processing operations we conduct on the OSN data, which are novel or adapted for use with our social media data. All of these are also available in an open-source pipeline framework for streaming and parallel processing introduced in [Preoţiuc-Pietro et al., 2012].

Chapter 4 has the role of motivating further research by showing that social media data is time dependent. First, we overview previous work which studies temporal influence or modelling, focusing on social media applications. We then analyse correlations with real world trends, in particular to political party voting intention, and show significant correlations with these, together with predictive power. Using supervised models that take time into account and place more emphasis on recent data we manage to improve the predictive accuracy.

Chapter 5 presents a supervised learning setup based on Gaussian Process regression for identification, categorisation and forecasting of temporal trends. We show that by using different kernels and performing Bayesian model selection, we can identify temporal trends that are more complex than linear or smoothly varying with a special emphasis on periodicities. This gives better results for forecasting of word behaviour when compared to a number of other methods. Furthermore, these results are proven to be useful downstream. By taking this prediction into account, time-aware text classification yields better results than the time-agnostic methods. We also study user behaviour uncovering similar patterns to words.

Chapter 6 proposes an unsupervised learning setup which aims to use the variations in word usage over time to uncover events. A method based on the distributional similarity assumption for words is formulated. This hypothesises that words in a restricted time frame with similar distributional proprieties are indicative of the same event. Based on co-occurrence information from short social media texts, we develop methods which cluster words together in a time frame. A comparative analysis is performed using the same method on data aggregated over a larger time span. A temporal method that relates these clusters is also developed which is important for highlighting topic drift.

Finally, Chapter 7 summarises the findings of the thesis and indicates directions of future work.

1.3 Published material

Parts of this thesis overlap with work published in peer-reviewed venues. This is as follows:

- The framework for social media data pre-processing presented in Chapter 3 was developed in collaboration with colleagues from the University of Southampton and first introduced in [Preoţiuc-Pietro et al., 2012].

- The user geolocation algorithm presented in Section 3.1.7 was developed in collaboration with my colleague Dominic Rout and was published in [Rout et al., 2013].
- Results from Section 4.3.3 are based on work in collaboration with Vasileios Lampos that were first introduced in [Lampos et al., 2013].
- The user behaviour data analysis in Section 2.2.3 and the venue category analysis from Section 5.3 were published in [Preoţiuc-Pietro and Cohn, 2013b].
- The models from Chapter 5 and the experiments relating to hashtags were original contributions published in [Preoţiuc-Pietro and Cohn, 2013a].
- The aggregate experiments on the Austrian Politics dataset from Chapter 6 are presented in [Preoţiuc-Pietro et al., 2013b].

Chapter 2

Online Social Networks

This thesis deals with Online Social Networks (OSN), a relatively new source of data. For this reason, it is important to define and understand what are OSNs, specifically the two types we consider further, and what are the peculiarities of data coming from these. In order to put our thesis into context, we aim to overview research on the properties of these networks. The most important OSNs, such as Twitter and Foursquare, have tens to hundreds of millions users worldwide. Another aim of the chapter is a description and analysis of the data we collect and use in the rest of this thesis.

The first two sections overview the two types of OSNs we consider in the rest of the thesis: microblogs in Section 2.1 and Location Based Social Networks (LBSNs) in Section 2.2. In each of their respective sections, we also introduce the datasets used in the rest of the thesis. Further, an analysis that presents the general properties of the data is conducted which aims to highlight the features that make OSN data of significant research interest compared to traditional data sources.

2.1 Microblogging

Microblogging platforms, the most important of which is Twitter, have evolved since 2006 as an easy-to-use alternative to blogging and have quickly matched these in popularity. Blogs (i.e. web logs) are usually maintained by individuals and they express subjective entries like commentaries, opinions, photos and videos. Posts are displayed in reverse chronological order. Blogs are usually interactive, allowing visitors to comment on posts or chat to each other. A basic element of social networking is also involved as blogs have a 'blogroll' in which other related blogs or blogs of friends of the owner are added. Blogs became popular in the early 2000s being part of the emerging Web 2.0 movement. In September 2013 the number of blogs is estimated

to be around 270 million across the 3 most popular platforms (Tumblr, Wordpress, Blogger). Even though blogs are important sources of data and have been studied by researchers [Facca and Lanzi, 2005], the lack of social interaction between them, as well as the diversity of the blogging platforms and the difficulty of data collection allowed only limited research progress.

Microblogs are blogs in which the post length is limited usually to a very low value. The most popular micro-blogging service is Twitter with a number of around 500 million registered users as of March 2013 out of which 200 million are active monthly.[1] Twitter is a OSN and microblogging service that allows its users to send and read other users updates (known as 'tweets'), which are text-based posts of up to 140 characters in length. The length of the message was restricted when the platform was created in 2006 especially so that users can post updates on Twitter via SMS messages from their phone. Even though most people now use mobile phones that enable mobile Internet access and posting through dedicated applications, the maximum message length was kept at the 140 character limit. This encourages conciseness and effectiveness in communication, which together with access mobility, play an important role in reducing the cost of information sharing and promoting timely updates. This brevity also makes users that receive the information to easily browse and monitor large numbers of users.

Like every OSN, Twitter allows users to connect with other users but, unlike most of other OSNs, this relationship is not mutual. If a user is interested in other users' posts they 'follow' that user with no need of their approval (except if the 'protect my tweets' option is turned on). The group of users that follow a user are called his 'followers' and the group of users someone follows are called their 'friends' or 'followees'. When a user sends a tweet in the stream, this is displayed on his profile page, in the stream of his followers and in the Twitter public timeline (subject to the 'Protect my tweets' option) so these messages can be viewed by those following, by those visiting the profile or by those who search the public timeline by keywords. Twitter offers very good integration with other OSNs. A Twitter account can be linked so that updates can be pushed from other OSNs (e.g. Foursquare) to the Twitter stream. The stream can be monitored for updates via the Twitter website, SMS, RSS or other widgets or third party applications.

Twitter became widely adopted because it's a very easy and convenient method of communication in which people can interact with thousands of others in just a

[1] `http://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats/`

few seconds no matter their location. Users can share information, ask questions, network, receive breaking news and follow your friends updates. From the start of the service, it has also put a great emphasis on mobile connectivity, with a large number of lightweight applications and support for mobile phones. As of March 2013, 60% of all Tweets come from mobile devices. If the mobile devices have their geolocation tracking option turned on, each tweet will have attached the co-ordinates from which it was sent. This represents a huge novel opportunity for researchers to gain insight from this geo-tagged data, although only around 1-2% of all tweets are geo-tagged.

2.1.1 Conventions

Twitter users have developed a set of conventions which we will briefly present:

@ reply In Twitter, users can directly send messages to a user by using the @ symbol before their user name. That message will be displayed to that users stream indicating who is communicating with them. The @ messages are also used in tweets in order to make a reference to other users or indicating intended recipients of messages that are posted in an otherwise public forum in order to gain the target persons attention, which is essential for conversation to occur. The convention is that in a direct message the @ mention is first in the tweet, while in a referencing message the @ appears in other parts of the tweet.

Honeycutt and Herring [2009] offer insights into the usage of @ replies. Sousa et al. [2010] investigates what motivates users' interactions. The study concludes that the social aspect appears to be predominant to the topical aspect. However, a slight tendency was observed for users with larger networks to separate their connections depending on the topics discussed.

Retweets Retweeting is the Twitter equivalent of email forwarding where users post tweets originally posted by others. This convention emerged from Twitter users. Due to this fact, unlike @replies and hashtags, the conventions for retweeting were, until recently, hugely inconsistent although 'RT @usr msg' has emerged as the common form of retweeting the message 'msg' tweeted by the user 'usr'. In 2010, retweeting has become an integral feature of Twitter although most users still tend to follow the old conventions.

Because Twitter limits tweets to 140 characters, messages being retweeted must frequently be modified to accommodate the additional notation meant to indicate

that the message is a retweet. While retweeting may simply be seen as the act of copying and rebroadcasting, most people regard this as an act of approval of their tweets value. For this reason, some of the most visible Twitter participants retweet others and look to be retweeted [Boyd et al., 2010].

Hashtags Hashtags represent crowdsourced free-form text labels that are part of tweets allowing users to assign them to a discussion thread and enable users to search for them. Due to the massive amounts of data that are published in the public Twitter stream there is a problem in organising for searching, browsing and making sense of all these messages. A particular tweet can have more than one hashtag attached to it. The hashtags can be considered a proxy for the topic because they group similar tweets [Laniado and Mika, 2010].

The practice of using keywords to label tweets most likely parallels the use of 'tags' to categorise web content. Tagging gained visibility with social bookmarking, but has expanded to all other forms of social media. Usually, hashtags are very Twitter specific and can be used as regular words integral to the tweet, referring to a general topic (#jobs) or emotion (#sadtweet), current events (#25jan) or Twitter memes (#backintheday) or games (#followfridays). Definitions of hashtags can be found at `http://tagdef.com/`.

Trending topics The large volume of Twitter content generates very different and diverse topics of interest. When topics or trends arise around the world, Twitter acts as filter and amplifier of these and thus contributes to the public agenda. In this competitive environment, the topics that manage to attract the largest attention in the system reaching the top in terms of popularity are called 'trending topics'. Although the exact algorithm of detecting the trending topics is kept secret by Twitter, studies have found that the content that trends is largely news from traditional media sources, which are then amplified by repeated retweets on Twitter to create trends [Asur et al., 2011].

For more details, a complete description of Twitter and its conventions is presented in [O'Reilly and Milstein, 2009, McFedries, 2010]. Due to a very large base of engaged users that post multiple tweets every day, the textual information and the directed structure of the network, Twitter has become by far the most used OSN in research studies. The relative ease of collecting data is also a key reason why Twitter is popular with the research community. The specific methods for collection we will use are presented in the next section.

2.1.2 Quantitative analysis

In this section we will briefly present the main quantitative characteristics of the Twitter microblogging service in order to better understand the context of our research. The around 500 million Twitter users that generate on average 500 million tweets per day (as of October 2012)[2] are not representative of the demographics of the entire population. A study on demographics of people using Twitter was conducted in the U.S.,[3] one of the top three most active countries with 18% of its entire population using the service. This study showed that younger people (aged 18-29) are significantly more represented than all the other older categories, with lower percentages as the age group increases. Also, people with a higher household income, more education and living in urban areas are more likely to use Twitter. Ethnicity also shows significant differences, with people from a hispanic or afro-american background being more attracted to using Twitter.

However, not all the users on Twitter use the system with the same frequency. Research [Wu et al., 2011] has concluded that from all the Twitter users, 50% of tweets are generated by 0.01% of 'elite' users. Heil and Piskorski [2009] drew a similar conclusion, reporting that just 10% of Twitter's users produced 90% of the tweets in 2009.

Researchers have studied why and how people use microblogging, providing different categorisations of users. In the first study on why people use Twitter, Java et al. [2007] identify the following reasons: conversations, sharing of daily activities, seeking and sharing information and reporting of current events. The users were in turn divided into sources of information, friends or information seekers, a distinction that was more or less acknowledged in future work. [Zhao and Rosson, 2009] identified that the main types of content users share on Twitter are personal whereabouts, links and opinions. The social conversational and collaborative aspect was studied in [Honeycutt and Herring, 2009] suggesting that Twitter can be used as a new informal communication service. [Jansen et al., 2009] analysed Twitter as 'electronic word-of-mouth' finding that around 19% of tweets talk about brands, out of which around 20% present an opinion relating to it. [Naaman et al., 2010] classified users in terms of content into 'informers' and 'meformers'. The former category of users represent a minority (around 20% at that time) which produce new content related to events external to themselves, while the later and larger category share content relating to them: their whereabouts, feelings, opinions or self-promotion messages.

[2] http://www.telegraph.co.uk/technology/twitter/9945505/Twitter-in-numbers.html
[3] http://pewinternet.org/Reports/2013/social-networking-sites.aspx

Another significant feature of Twitter is its social network of followers. A large scale analysis that crawled the entire social graph of Twitter as of June 2009 (only 41 million users back then) is presented in [Kwak et al., 2010]. In the directed graph topology analysis the authors have found a non-power-law follower distribution, a short effective diameter of the graph, and low reciprocity. This contradicts the social network remarks of Java et al. [2007] showing a maturation of the OSN and a deviation from known characteristics of human social networks.

Kwak et al. [2010] analysed and reported tweets in top trending topics based on their temporal behaviour and user participation. The trending topics were classified based on the active period and show that the majority of topics are headline news or persistent news. This shows that Twitter acts as a filtering and dissemination platform for real world news and events. Quantitative information about information diffusion shows that diffusion of news in Twitter is almost instantaneous. Another finding of the study is the existence of a two-step flow of information. Almost half the information that originates from the media passes to the masses indirectly via a diffuse intermediate layer of opinion leaders, who although classified as ordinary users are more connected and more exposed to the media than their followers. All in all, we can draw the conclusion that Twitter is more of a broadcast channel.

Moreover, internal Twitter statistics have shown that 40% of users worldwide simply use the service as a 'curated news feed of updates that reflect their passions' with many accounts having posted no tweets at all.

2.1.3 Data

The data we collect for our experiments is obtained from the public APIs that the Twitter microblogging platform offers for free. The data provided by the APIs is presented to the end-user into the popular lightweight JSON format. These APIs are restricted in the data they provide and the rate with which to collect it. Privacy restrictions are enforced by the OSNs that own the data. They claim the rights for that data and do not want it fully accessible in the public domain. An agreement by the Library of Congress will offer an improvement in data access in the near future.[4] Considering the huge volume as another factor, the distribution of data sets has thus become a real issue. The replicability of research is a hot and fiercely debated topic in the scientific community.

[4] `http://blogs.loc.gov/loc/2013/01/update-on-the-twitter-archive-at-the-library-of-congress/`

One of the workarounds is anonymising parts of the data so that user information is not viewable or deductible. Also publishing the aggregate statistics and feature representations we will feed to our models is a way of anonimisation. Another alternative we use for publishing datasets is by using the method introduced as part of the TREC Microblog task. The data is distributed as a list of unique tweet and user id pairs. Using a PERL script any user can re-crawl the data from the website on their own machine, although deleted tweets will become inaccessible.

This section provides a description of three datasets we will use throughout the rest of the thesis. The collection code developed for downloading the data is published open-source.[5] An extensive coverage of the Twitter APIs is presented in [Makice, 2009].

2.1.3.1 Gardenhose dataset

One of the two public Twitter APIs is the Streaming API[6] which feeds large amounts of real time data to users. However, rate limiting constraints are enforced[7] and the public timeline only constitutes a statistical representative sample of the entire data.

The method is very popular for getting hold of large volumes of live data which can be later filtered. We have used a local server to gather data using the Twitter Streaming API since 2009 and, with very little gaps, have collected data up to the time of writing. The access level we posses, Gardenhose, allows storing of a 10% sample of the entire stream. The size of the dataset (in GB) over time is presented in Figure 2.1. The data in compressed using the LZO (LempelZivOberhumer) block-compression algorithm.[8] The compression rate is around 19 with 1GB compressed data holding around 7 million tweets. The general trend in Twitter data is to increase over time. The lower dataset size at the end of 2011 and beginning of 2012 is due to only receiving 1% of the sample from Twitter during that time. A few logistic issues over 2013 have resulted in a slight loss of data in some months.

2.1.3.2 U.K. Users dataset

When we want to analyse specific user behaviour, the Gardenhose stream is usually not good enough because it only presents a sample which possibly misses important information. We have thus developed a collection scheme focused on specific sets of

[5] `http://github.com/danielpreotiuc/twitter-collection-utils`
[6] `http://dev.twitter.com/docs/streaming-apis`
[7] `http://dev.twitter.com/pages/rate-limiting`
[8] `http://www.lzop.org/`

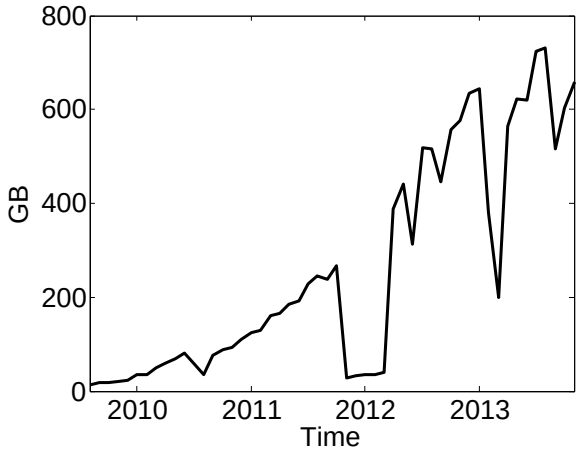


Figure 2.1: Size of the Gardenhose dataset over collection time.

users. This scheme uses the other public Twitter API, the REST API[9] (or Search API), to cycle through a set of users and retrieve the new tweets with regularity. The REST API gives access to the entire Twitter stream, but it imposes rate limits and restrictions for retrieving historical data.

The dataset in this section is collected using this method and aims to provide a general snapshot of active Twitter users in the United Kingdom. We have collected all users that authored tweets in the Gardenhose stream during one month. We then assumed each user to be from the U.K. if the location field in their profile was matched with a list of all U.K. cities (detailed in Section 3.1.7) and their timezone was set to G.M.T. In this way, we extracted hundreds of thousands of U.K. users. In Figure 2.2 we presented the distribution of users across the U.K. which shows a similarity to the distribution of the actual population.

Although the data collection is a continuous process, for our experiments we use data authored between 30 April 2010 and 13 February 2012. This spans 2 years in which many important events took place in the U.K., for example the general elections (May 2010) or the riots (August 2011). Complete dataset statistics are given in Table 2.1.

No.tweets	**No.users**	**Size**
6,655,146	42,484	16.6 GB

Table 2.1: Statistics for the U.K. Users Dataset.

[9] `http://dev.twitter.com/docs/api/1.1`

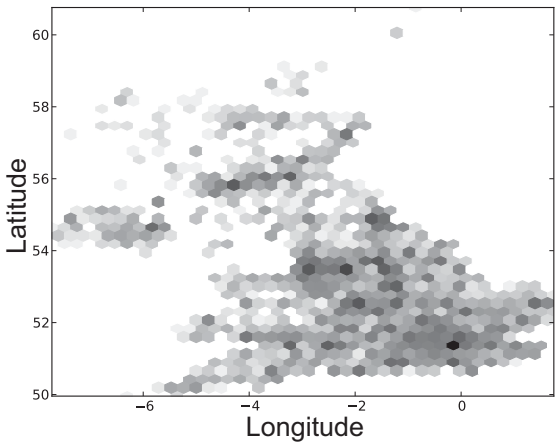


Figure 2.2: Distribution of the U.K. Users.

2.1.3.3 Austrian Politics dataset

The other dataset we use is collected by the same user-centric method as the previous one. It aims to present a snapshot of the users that are talking about the same broad topic (i.e. politics) in a country. For this, we called upon the expertise of SORA,[10] an Austrian institute for social research, to provide us with an extensive list of Twitter users that discuss politics and current affairs, with an emphasis on Austria. The dataset spans the period of 25 January 2012 to 31 August 2013, although as in the previous case, the dataset is continuously updated. This is slightly smaller size and length than the previous one, but we expect it to be *cleaner* in terms of content, with fewer conversational, spam or self-promotion messages. The dataset statistics are presented in Table 2.2.

The dataset is also particularly interesting because most of its contents is written in German. This is helpful for establishing the portability of the methods we describe in this thesis.

No.tweets	**No.users**	**Size**
2,585,364	1,973	5.7 GB

Table 2.2: Dataset statistics for the Austrian Politics Dataset.

2.2 Location Based Social Networks

With the widespread adoption of smart-phones and mobile Internet, users have started sharing information about their location with their friends. Location is increasingly

[10] `http://www.sora.at`

becoming a crucial aspect of many online services: people appear more willing to share their geographic location with friends, giving the opportunity for companies to customise their services by taking into account where the user is located. Location is assumed to be highly dependent on time: based on the time of day, day of week or holidays, users prefer specific locations. It is thus expected that models using location data should also integrate temporal information.

The first OSNs based on location sharing appeared in 2009 and they have by now reached maturity. The most well known Location Based Social Network (LBSN) is Foursquare, with other popular OSNs such as Google+ also having location sharing features. Although these networks lack a widespread audience the size of other OSNs, their target audience is formed of regular and devoted users of the service. Statistics from November 2013 show Foursquare having around 40 million users.[11]

The main functionality of LSBNs is that they allow users to share their current location. This location, corresponding to the geo-coordinates of the user, is focused on venues. Venues are user-specified real-world places with geo-spatial coordinates. The sharing of location in a predetermined venue is called a *check-in*. A user can add tips, categories, tags, comments and upload photos or videos regarding once he checkes-in to an actual venue. In this way, places are semantically enriched with collaborative user knowledge. It is estimated that 1.5 million venues are annotated in the Foursquare platform.[12] The user can also see information about people who are nearby or who have been there before. These check-ins can be further pushed to other more mainstream OSNs such as Twitter and Facebook. As an addition to other LBSNs, Foursquare provides game features like mayorships and badges which encourage regular use of the system.

As in normal OSNs, LSBN users can make friends with each other in order to monitor their friends' location, tips and status. The friendship relationship is mutual, requiring each user to accept friendship requests to allow location sharing. The data arising from this application is subject to privacy constraints. The check-in data for users is thus not directly available to any user. Only the friends of an user have access to only recent locations, making data collection on a large scale challenging.

2.2.1 Quantitative analysis

In this section we will briefly review the most important properties of LBSNs as identified by other quantitative studies. With respect to Foursquare, geographic user

[11] `http://foursquare.com/about`
[12] `http://business.foursquare.com/`

activity patterns are studied in [Noulas et al., 2011]. The study on around 12 million check-ins analyses user check-in dynamics, demonstrating how it reveals meaningful spatio-temporal patterns and offers the opportunity to study both user mobility and urban spaces. The geo-temporal dynamics of collective user activity on Foursquare shows how check-ins provide a means to uncover human daily and weekly patterns, urban neighbourhood properties and recurrent transitions between different activities.

Data about the interplay between users and locations are for the first time available to researchers by analysing LBSN data. Analysing socio-spatial proprieties provides unprecedented chances to understand how users actively engage with places and online friends. [Scellato and Mascolo, 2011] quantitatively analyses user activity in Gowalla, a now defunct LBSN. The Gowalla API allowed access to the social network to obtain the dynamics with which users add new friends. The study highlights the differences in the distributions of friends, check-ins and places of Gowalla users. The temporal evolution of these distributions is analysed, noting that users add new friends at a faster rate than they accumulate new places and check-ins and presenting how these variations may influence the aggregated statistics. Scellato et al. [2011b] analyse the network of friends of a user together with their geographic location using data from three LBSNs (i.e. Foursquare, Gowalla and Brightkite). The study highlights how observed properties deviate from what would be expected by chance with purely social or geographic mechanisms. Users exhibit different characteristic geographic scales of online interaction, with weak positive correlation between number of friends and their average distance. Also, a similar heterogeneity appears with respect to social triads (i.e. groups of three interconnected users), with users participating in geographically wider triangles as their number of friends increase.

The trails of user locations (mobility patterns) are also revealed by LBSN check-ins. A comprehensive quantitative study using Foursquare data was conducted by Cheng et al. [2011]. The authors study human mobility patterns and explore factors that influence this mobility, including social status, sentiment, and geographic constraints. As data, the authors used over 22 million check-ins across 220,000 users. They analysed the spatial, temporal, social, and textual aspects associated with these check-ins. The authors concluded that LBSN users follow the Levy Flight [Rhee et al., 2011] mobility pattern and adopt periodic behaviours. The users mobility patterns seem related to their geographic, economic constraints as well as their social status. A basic sentiment analysis regarding the comments or tips associated with the check-ins was also provided, showing that this data could be exploited in work that combines the multiple modalities of the data.

Initially, mobility patterns were studied using proxies to movement such as U.S. banknote movement [Brockmann et al., 2006] and marine predators [Sims et al., 2008]. The development and availability of portable devices like mobile phones made tracking of people's location easier. For example, Gonzalez et al. [2008] analysed a dataset from 100,000 users over 6 months. This dataset contained the locations of the user's closest mobile phone tower every time they made a phone call. This allowed the researchers to have approximate data of each users' location.

2.2.2 Data

In order to study individual human behaviour we need access to the full data generated by users. This is due to sparsity of check-ins (only a few on average every day) and in order to reliably analyse transitions between locations. Until recently, the main bottleneck for this type of work was data availability largely as a consequence of concerns about anonymity and personal security.

On Foursquare, recent statistics[13] show a rate of around 6 million daily check-ins. Considering there are about 40 million users, we find that every user uses the system far less than once a day. Based on this, we conclude that we need to identify users that use the social network on a frequent daily basis.

2.2.2.1 Frequent Users dataset

Only the friends from the social network of a user have access to (a limited set of) his check-ins. In this section we present a novel collection method that overcomes this restriction and extracts all check-ins from specific users, using a combination of the Twitter and Foursquare APIs. This targets the users who choose to push their Foursquare check-ins to the public Twitter stream (around 20-25% of the total users [Noulas et al., 2011]). This way, their activity becomes available for collection. However, this is not a straight-forward task as the venue data is available from the Foursquare check-in page linked to from the tweet. This requires crawling and parsing in order to collect the venue information and then linking it back to the original timestamp and user data collected via Twitter.

We name this dataset 'Frequent Users Dataset' and we define a 'frequent user' as one that uses Foursquare at least 3 times per day. We identify a number of 'frequent users' of Foursquare using the Twitter Streaming API. The dataset collection interval is 31 August 2011 – 1 October 2011. The dataset statistics are presented in Table 2.3.

[13] `http://go.bloomberg.com/tech-deals/2013-04-11-checking-into-whats-behind-foursquares-41-million-infusion/`

No.users	No.check-ins	No.check-ins/user
9167	959,122	104.6 ± 49.4

Table 2.3: Frequent Users Dataset statistics

The average number of check-ins per user and day is more than 10 times the number presented in other studies [Lian and Xie, 2011, Cheng et al., 2011] which collect Foursquare check-ins via Twitter in bulk. Cheng et al. [2011] specify that 72% of the users have fewer than 100 check-ins in a 5 month interval. When performing an aggregate analysis over the check-ins a sample of the data is sufficient. However, performing a study on individual user patterns is not possible if using a random sample of the data (e.g. transitions between consecutive venues will be altered) but only if using the full data of frequent users.

Our approach considers only users that choose to push their location sharing information on other public social networks, specifically Twitter. We note that this set of users may not be representative for the entire Foursquare user base or for the population in general, but it will still span multiple types of users that differ in behaviour, location and age group. In the next section we conduct an analysis in order to determine if our dataset presents the same characteristics in terms of group behaviour as other studies of Foursquare or other sources.

The collection code developed for this is available as open-source software.[14] The dataset is available in an anonymised format using the TREC Microblog method.[15]

2.2.3 Data analysis

In this section we will study the proprieties of our LBSN data and relate the dataset statistics to previous findings in human mobility patterns study e.g. [Gonzalez et al., 2008]. Our aim will be to show that the datasets have similar proprieties even if they were extracted from different sources.

Time distribution We start by analysing the time distribution of check-ins. A plot of the number of check-ins for the entire dataset is presented in Figure 2.3.

As our data is collected during a month, we could expect to find regularities in the size and time of the total number of check-ins. We observe that, in general, there is a consistent weekly pattern of activity, with weekdays each having three peaks

[14] `http://github.com/danielpreotiuc/foursquare-collection-utils`
[15] `http://www.dcs.shef.ac.uk/~daniel/foursquare/`

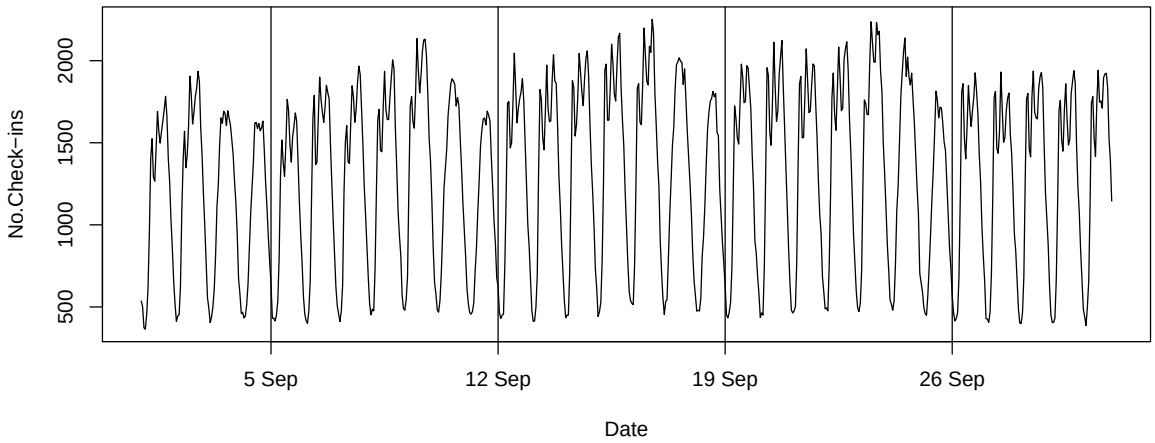


Figure 2.3: Time distribution of check-ins over a month. The start of the week is indicated by vertical lines.

during the day and weekends having a smoother distribution and generally lower activity. Transitions between days can be observed by very low activity during nights as expected. We also notice a slightly upward trend in the number of check-ins in weekdays as the week progresses.

We plot in Figure 2.4 the check-in frequency per each weekday aggregated over the four fully observed weeks in our dataset. We notice the same daily period of check-ins that spikes at 3 times: at 9am when people start their day, at noon when most of the people have their lunch break and in the evenings when they leave the workplace or go out. Here we observe a trend: whilst check-ins in the morning and afternoon are similar, we notice an increase in activity in the evenings from Monday to Friday. This reveals that as the week progresses and it gets closer to the weekend, people tend to visit more places in the evenings.

Gyration We now examine the radius of gyration for each user. This will give us an indication of how often a user travels far away from his *home* location. The radius of gyration for a user is computed according to the following formula:

$$g = \sqrt{\frac{1}{n}\sum_{i=1}^{n} dist(loc_i, loc_{home})} \tag{2.1}$$

The *home* location is chosen as the most frequent venue the user checks-in to. As the measure of distance we consider the great-circle distance to account for the Earth's curvature. The distribution of radius of gyration is presented in Figure 2.5. A low radius of gyration indicates users that travel only locally, while a higher radius of gyration indicates users that often travel long distances.

By analysing the graph, we can see that the distribution roughly follows a power law of the form $x^{-\beta}$ with $\beta = 1.45$ (if ignoring the first points of the distribution). Comparing our results to those in [Cheng et al., 2011] we find out that our distribution is much more compact. As a comparison, we only have 4.3% of users with a gyration than more 500 miles, while Cheng et al. [2011] report 14.6%. This is due to the fact that we consider only frequent users that check-in at frequent time intervals, as opposed to the sampling approach used by Cheng et al. [2011] that uses all users and only a subset of their data. Another factor is that infrequent users check-in rarely and have a higher possibility of taking large trips between these check-ins.

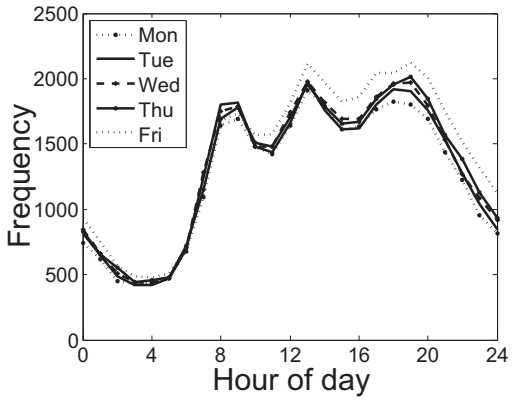


Figure 2.4: Weekly check-in frequencies.

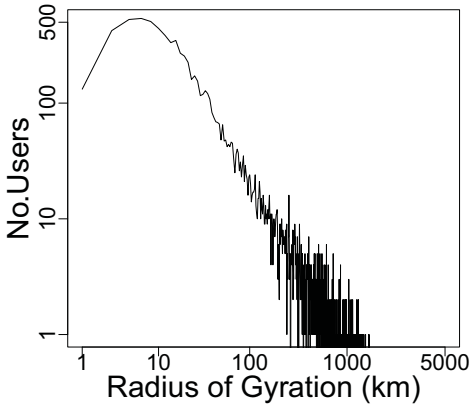


Figure 2.5: Distribution of radius of gyration (km).

Interevent times and distances Further, we investigate the proprieties of consecutive check-ins of the same user. We study both the interevent times and distance. The distribution of the former is presented in Figure 2.6 while the latter is presented in Figure 2.7, both plotted in log-log space.

The distribution of interevent times indicate that most of the check-ins are performed within a 2 hour interval from the previous. This is because most check-ins in our dataset are performed at transport hubs, shops or food outlets where people do not spend much time. The frequencies decrease with increasing time intervals, with the exception of a small plateau at around 8-10 hours. This represents the usual length of the workday. The interevent distance fits a power law of the form $x^{-1.56}$, with very few consecutive check-ins more than 100 km from each other.

Venue frequency distribution We now study the distribution of check-ins/venue for users. This is for each user the distribution of frequencies for every venue that he

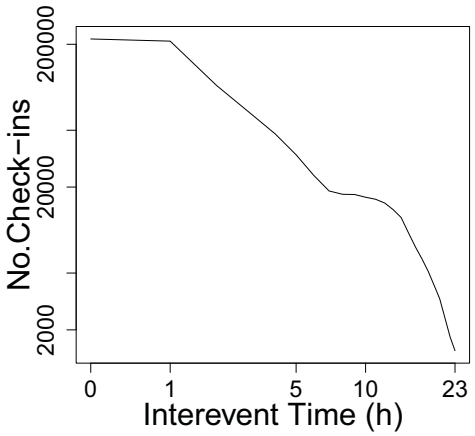


Figure 2.6: Distribution of interevent times (h).

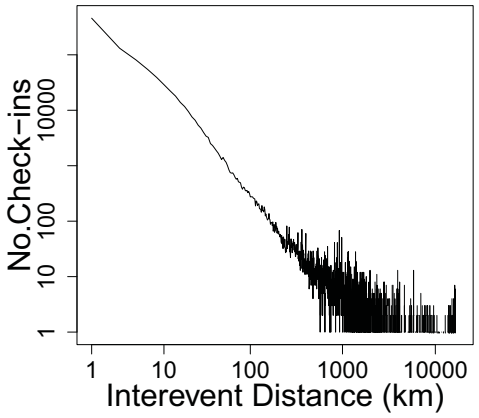


Figure 2.7: Distribution of interevent distances (km).

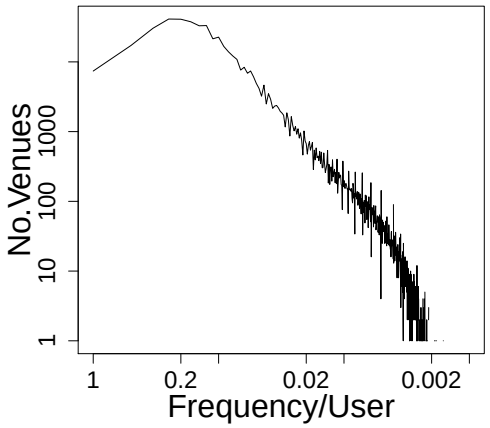


Figure 2.8: Venue Frequency Distribution.

has checked in. We aggregate all the distributions corresponding to every user in our dataset and show the results in log-log space in Figure 2.8.

Most users have a few places they visit very often (the peak at 10-20% of their total check-ins) and many places they seldom visit. It has to be noted that while it is likely that users would get 'bored' of performing check-ins in the same place they visit frequently, the game elements of Foursquare which offers mayorships keeps users motivated to continue checking in.

Returning frequency Now, we examine the periodic patterns in human mobility by measuring the returning frequency at a venue. The returning frequency is the number of times a user returning to a place visited h hours ago. This is presented for our dataset in Figure 2.9.

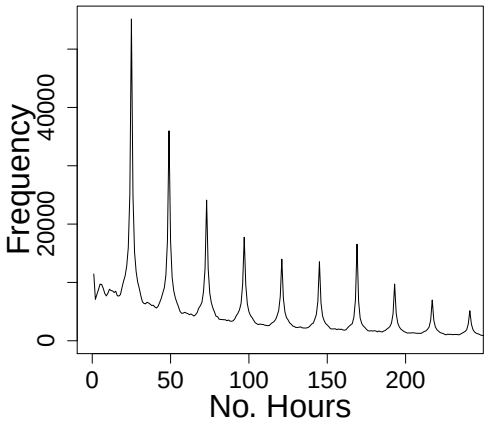


Figure 2.9: Returning frequency.

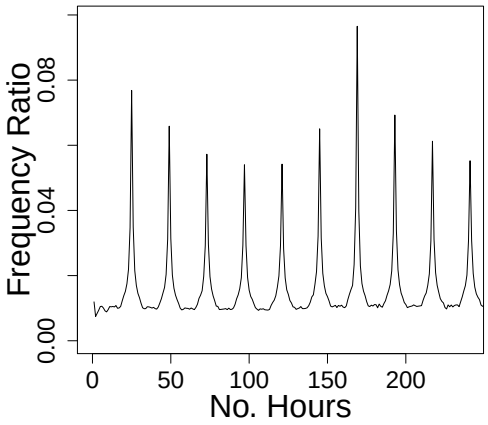


Figure 2.10: Returning frequency ratio.

We notice spikes for daily intervals, which diminish rapidly for the future days. We also notice a strong weekly (168 hours) return probability. This behaviour is similar to the one reported in [Gonzalez et al., 2008] who tracked mobile phone positioning. However, when comparing it to a similar figure from [Cheng et al., 2011] that uses the same data source as us, we observe that the downward trend in the returning probability is diminished in their dataset. We can explain this by the fact that the authors used a sample of the underlying distribution. The returning probability gets 'spread' in future days, smoothing the distribution.

We also compute a *returning frequency ratio*. This measures the number of times a user returned to a venue after h hours divided by the times he was observed after h hours. Because we consider only days when users register more than 3 check-ins, we exclude these days and adjust the ratios accordingly. This technique corrects for sporadic or bursts use of the LBSN such that longer term patterns can be more reliably identified. The distribution of returning frequency ratios is presented in Figure 2.10.

From this figure we observe that the highest returning ratio is the weekly one. High values are obtained for daily returning ratios as well. This shows that it is more likely that one visits the same place as a week ago than a place that one visited the day before. A reason for that may be the different activities one does on weekdays and weekends which break the daily pattern.

2.3 Conclusion

In this chapter we have presented an introduction to OSNs, focusing on two specific networks: one based on sharing text (Twitter) and another on location (Foursquare). We have placed a specific emphasis on the different novel properties of data arising

from OSNs compared to other sources. The aspects include the social and the conversational, in the form of the underlying social network structure, the spatial and the temporal dimensions. For example, a deeper analysis on mobility patterns in Foursquare has shown strong temporal effects (e.g. periodicities) in user activity and types of places visited.

Data collection from OSNs can be challenging, mainly due to traffic limiting and privacy constraints, although it is provided for free to consumers. We have presented open-source libraries for data collection and introduced the datasets that will be used for the rest of the thesis. Also described was an analysis of these datasets, including their temporal characteristics.

As discussed within the chapter, several challenges arise with the study of OSN data. From an engineering perspective, the sheer volume of data is big and simply storing, loading in memory or accessing all the data leads to many issues and a need for new tools to be used. From an algorithmic point of view, the main issue we face is again the volume of data with most of it being irrelevant for our goals. When analysing text however, even simple text processing operations, like tokenisation, need to be rethought. Being of short length, many texts from OSNs exhibit non-standard language and capitalisation use, abbreviations, new terms or short-lived conventions while also suffering from lack of context. An architecture for processing this data in order to be used by further algorithms or applications will be the topic of Chapter 3.

Chapter 3

Architecture for experimental setup

The ability to create representations of data is an essential part of designing applications and algorithms. As we have seen from the previous chapter, there are peculiarities with OSN data that do not allow us to use well established NLP tools for data processing. In this chapter we aim to present a viable architecture that addresses this problem. This contains multiple modes, each performing a specific task, in a pipeline framework that can be scaled to large volumes of data and used in an online setting.

We describe each individual component for data pre-processing in Section 3.1 where we also present the results of a novel algorithm for identifying user location based on their social ties. The overall open-source framework which incorporates all these modes is described in Section 3.2, together with the design aims, architecture and data format.

3.1 Components

We first describe the individual components of the framework for OSN data pre-processing. Usually (e.g. on newswire data) this is performed by standard NLP tools developed over time by the NLP community. However, like previously highlighted, data (and text in particular) in OSNs has different properties to the one normally used in NLP tasks. OSN data is usually noisy and contains many service specific conventions, misspellings, shortenings, abbreviations, the texts have short length and suffer from ambiguities due to restricted context. For this reason, specific tools have to be created or adapted to the peculiarities of OSN data. We identified a number of tools that are useful for our downstream algorithms and implemented each as an

individual component which can use input from previous modes. We note that some of the described modes are not used in the rest of the thesis, i.e. the stemmer from Section 3.1.3 and the sentiment detector from Section 3.1.6.

3.1.1 Tokenisation

Tokenisation is the process by which an unstructured input text it split into lexical units (called tokens) that carry a meaning of their own, like words, punctuation marks or emoticons. This process stands at the foundation of any subsequent NLP applications. Usually this is performed by a standard set of rules (e.g. white-space delimitation), but OSN data introduces a few extra challenges. In particular, we need to correctly identify and handle tokens such as URLs, sequences of punctuation marks, emoticons, Twitter conventions, abbreviations and dates, not all of which have orthodox forms (e.g. no white-space between punctuation and words, URLs with no leading *http*).

Our implementation works through a chainable set of regular expressions which define patterns of tokens that it aims to 'protect', followed by white-space delimitation of the remaining tokens. Further different categories of tokens to protect can be specified. In this way, we create classes of tokens: 'protected' (e.g. OSN conventions, emoticons, URLs and punctuation) and 'unprotected' (e.g. regular words and numbers) together with a combined set of both categories tokens.

Examples of tweets and their tokens as extracted by our method are in Table 3.1. Note that the tokenisation scheme only works with Latin scripted languages and their conventions (e.g. delimitation between words by punctuation or white-space). Other languages (e.g. Chinese, Hindi) pose fundamentally different issues for tokenisation [Wu and Fung, 1994] and have not been analysed.

Tweet	RT @USER: Ah!Had d most amazing meal,Love WASABI at th Taj,defntly d best japanese food in India:)had 2 loosen my belt ate so ...
Tokens	[RT, @USER, :, Ah, !, Had, d, most, amazing, meal, ',', Love, WASABI, at, th, Taj, ',', defntly, d, best, japanese, food, in, India, :), had, 2, loosen, my, belt, ate, so, ...]
Tweet	"ALL JUMPERS/MINI PRIX HAVE BEEN CANCELED FOR MAY. NEW SCHEDULE/START TIMES AVAILABLE AT www.farandawayfarmhorseshows.com"
Tokens	[ALL, JUMPERS, /, MINI, PRIX, HAVE, BEEN, CANCELED, FOR, MAY, ., NEW, SCHEDULE, /, START, TIMES, AVAILABLE, AT, www.farandawayfarmhorseshows.com]

Table 3.1: Tokenisation examples.

3.1.2 Language detector

Language detection is the task that detects the language of a text based on its linguistic content. Multiple systems for solving this task are readily available and have shown to perform very well on long texts [Cavnar and Trenkle, 1994]. Most language detection tools exploit character n-gram language models trained over large corpora for each language and then identify the most probable language under the models. The main challenge when applied on OSN data is the short number of tokens and their noisy nature. Baldwin and Lui [2010] identify a decrease in performance from 90-95% to around 70% with then state-of-the-art language detection algorithms when restricting the input text's length.

We use the language detection method presented in Lui and Baldwin [2011] which we have reimplemented in Java in order to fit into our pipeline. It uses as features a mixture of byte n-grams selected using Information Gain to be informative across all languages whilst not domain specific. These features are used in a Naïve Bayes classifier for assigning the most probable language to a piece of text. As input, we provide only the 'unprotected' tokens as determined by our tokeniser. Due to the short length of the text, we can make the simplifying assumption of one language per tweet. We choose the tool of Lui and Baldwin [2011] over others [Cavnar and Trenkle, 1994] for the following reasons: it is reported as being the fastest, it is standalone and is pre-trained on 97 languages, it works at a character level without using the script information (thus needing only the 'text' field as input).

We evaluate the method on the microblog dataset introduced by Carter et al. [2013]. Our accuracy on this dataset is 89.3% when using a 97-way classifier compared to 89.5% reported by Carter et al. [2013] but when using only a 5-way TextCat classifier [Cavnar and Trenkle, 1994]. Given that we aim to identify as many languages as possible, we conclude that the language identification performs sufficiently well. Further evidence on the suitability of this language detection method on OSN data is provided in [Lui and Baldwin, 2012].

Room for improvement is still possible in a few ways. First, Carter et al. [2013] suggest using other OSN specific information (e.g. the user self specified language) to boost accuracy to up to 97%. Further, there have recently been studies that go beyond the 1 language/text restriction in order to identify language switching inside a text [King and Abney, 2013, Nguyen and Dogruoz, 2013], but they are subject to scalability concerns.

3.1.3 Stemmer

We use the traditional Porter Stemmer, which represents the standard stemmer for NLP and Information Retrieval. We use the Snowball stemmer backed by the Terrier Snowball stemmer implementation.[1]

3.1.4 Local time

The timestamps of the OSN items are presented in time reported in G.M.T. However, for some applications such as human mobility patterns, we are more interested in the local time of the OSN data. To address this, we compute a local time for each item by simply adding the provided time offset relative to G.M.T.

3.1.5 Deduplicator

A common occurrence in OSNs is that of duplicate or near-duplicate messages. These can be caused either by redistribution of messages (e.g. the retweet convention in Twitter), automated messages (e.g. application updates) or spam accounts. Examples of duplicate tweets are presented in Table 3.2. When computing aggregate statistics over the data, such as co-occurrence counts, messages with many copies will bias these statistics, artificially increasing them and perhaps leading to wrong conclusions downstream.

Tweet	Sitting in green room with Justin Bieber...must resist urge to roundhouse kick him in his midget face
RT (not flagged)	Sitting in green room with Justin Bieber...must resist urge to roundhouse kick him in his midget face (via @WillFerrell) He he he
Added comment	He he he
Tweet	High hopes for new pneumonia jab: A new vaccine against pneumonia is being rolled out in Africa which, exper... http://bbc.in/gdRUfj BBC
Near-duplicate	High hopes for new pneumonia jab: A new vaccine against pneumonia is being rolled out in Africa which, experts s... http://bbc.in/gwn5Ir
Tweet	I just ousted @beamena as the mayor of Future bistro 483 bloor st west on @foursquare! http://4sq.com/aupeLA
Near-duplicate automated tweet	I just ousted @ryansullivan as the mayor of Noodles & Company on @foursquare! http://4sq.com/5ZHqvT

Table 3.2: Deduplication examples.

First, in order to correctly handle Twitter retweets not performed using the 'retweet' action (see Table 3.2), we have developed a set of regular expressions using different websites for Twitter etiquette (e.g. tweets starting with 'RT', ending with

[1] `http://terrier.org/`

'via @username'). Using these, we can identify if a retweet was performed and split it into the retweeted text and the comment that might have been added to the original message.

For identifying documents with duplicate or near-duplicates content within a set, we use the following strategy. Given that usually we want to deduplicate a large set of documents, we use the Bloom filter [Bloom, 1970] as a data structure for keeping track of existing content. The Bloom filter is a randomised data structure that indicates if an element is a member of a set. The data structure is space efficient and guarantees not to return false negatives, with the cost of returning false positives at a defined rate (e.g. 0.01 in our default setting). An analysis on the type of automated messages suggests that for near-duplicate detection it is sufficient to use the first 5 'unprotected' tokens (see Section 3.1.1). We thus ignore in deduplication the usernames, URLs and altered endings. For example, the last example in Table 3.2 represent frequent automated messages pushed from Foursquare about mayorship changes having different usernames and venues. We have thus multiple levels of deduplication, depending on how many of the following categories of documents we remove: duplicate content, near-duplicate content, retweets, retweets without comments.

3.1.6 Sentiment detector

Sentiment detection, by which the subjectivity and polarity of a document is extracted, is still a difficult task for automated methods. A special emphasis in recent research was regarding sentiment detection in OSN data, with properties of this data posing extra challenges to the task on a document level [Nakov et al., 2013].

However, for extracting aggregate opinions, simplistic methods that just rely on dictionary based sentiment detection appear to work sufficiently well in some downstream tasks [O'Connor et al., 2010]. We implemented a simple sentiment detector that computes a score for each document based on counting the words that belong to sentiment dictionaries, one for each polarity (positive and negative). The dictionaries we use are from OpinionFinder [Wilson et al., 2005]. The 'sentiment score' for each document is a number in the $[-1, 1]$ interval computed using the formula: $s = \frac{p-n}{p+n}$, where p and n are the number of 'positive' and 'negative' terms in the message. Examples of texts and their scores are shown in Table 3.3. The 'sentiment score' can then be generalised to datasets by computing the average of the individual documents sentiment scores.

Text	p	n	s
So im now at my dad's apartment and he gives me some ital food to eat..this shit look so nasty but taste really good.lol	5	2	0.71
Dear @barrackobama I seen you down in la. For dat oil spill is you comin here cuz I think this worse than oil an dead shrimp #imjustsaying	2	3	-0.2

Table 3.3: Sentiment detection examples.

3.1.7 Geolocator

A geolocator assigns to each OSN account a *home* location as the location from which the account emits most of its data from. A home location is represented as the pair of its geo-coordinates (i.e. latitude and longitude) which can be either an exact point on the map or a representative for a larger area (e.g. centre of a city or country). This is used in this thesis for identifying users in order to build the datasets presented in Section 2.1.3. Generally, there are four methods by which an account is assigned to a specific location: using the IP address, using its geo-tagged documents, the self-specified free text location description or automated methods that aim to infer it from data.

IP address geolocation achieves very good performance, with 70% of IPs correctly assigned within a 40km distance in the U.K.[2] However, access to an account's IP address is not available for regular applications and users.

The geo-tagged documents approach gives the most reliable estimate of a user's location, usually down to the exact geo-coordinates. If a user uses the OSN on a GPS enabled device and has the geo-tag feature enabled, the meta-data of the social media item will contain the geo-coordinates of the user when posting the message. The home location can then be selected as the most frequent pair in the user's recent or entire history. However, we have approximated using our data that only 1.7% of tweets are geo-tagged, which indicates that alternative methods are needed for better coverage.

Usually, each account on an OSN has profile information generated by the user which describes his location. Hecht et al. [2011] have shown that in the majority of cases, users on Twitter do not disclose information that can be used to accurately locate them, either by leaving the information blank, by providing vague information or a completely fictional description. Cheng et al. [2010] identify that 26% of users supply an actual city or town in this field. Examples of location field entries are presented in Table 3.4.

[2] `http://www.maxmind.com/en/city_accuracy`

Type	Location Field
Exact location	Manchester, UK
Ambiguous location	Newcastle
Vague location	United Kingdom
Non-Geographic entry	A blue telephone box

Table 3.4: Location field examples.

In order to geolocate accounts in the U.K. using the location field, we have compiled a gazeteer of U.K. city names and additional meta-data from DBpedia (e.g. geo-coordinates of the centre, region, population).[3] In total, the list comprises of 17,521 settlements. Because we will use the locations for training an algorithm, we aimed to develop a high precision method. This is based on regular expressions that looks up the content of the location field, extracts the city name and tries to match it with the gazeteer entries. The geo-coordinates of an account will be the centre coordinates of the city. A problem when using this method is the ambiguity between locations with different names (e.g. Newcastle in Tyneside and in Shropshire). In this case, in the absence of other context, the city with the largest population (according to DBpedia) is selected. While disambiguation within the U.K. is not a major issue, when considering locations in the entire world there might be many candidates for each location [Hoffart et al., 2011].

We have evaluated the accuracy of this method on a random sample of 1,000 Twitter accounts that were matched to a location by our method. Using two independent human annotators, we have found that 97,2% of the accounts matched the human judgements. Extensions can go beyond the U.K. and also look at alternative name variants for places (e.g. LDN for London). Still, the majority of accounts (65.63%) can not be matched to a location. For these, automated methods use machine learning with features devised from account data are used.

Automated methods can be divided into approaches that use the social network information, the textual content on the messages or both [Ren et al., 2012]. For using text, researchers have framed this problem either as a generative model [Eisenstein et al., 2010] in which the location of a user influences the words they use or as a classification task where text was used to create features for standard classifiers [Cheng et al., 2010, Mahmud et al., 2012]. Algorithms that explore the social connections of an account exploit the fact that users make connections often caused by real world interactions, which are still dominated by closeness in location [Liben-Nowell et al.,

[3] `http://dbpedia.org`

2005]. Backstrom et al. [2010] created a model for the distribution of distance between pairs of friends and used this to find the most probable location for an account.

In [Rout et al., 2013], we build a classification approach to geolocating users on Twitter and experiment on U.K. data. Using as ground truth the accounts mapped using the location field based method, we aim to classify user accounts into cities using as features the location of (a part of) his network connections, the city size, the reciprocated friendships and the triads in the social network. The performance for different geolocation methods is presented in Table 3.5 as the percentage of correct user-location assignments within a certain radius (in miles). We only consider as candidates the cities of the friends of a user, thus setting an upper-bound on the accuracy (denoted 'Oracle performance'). A 'random' baseline chooses at random only from within this set of locations. We also compare with the previous social network based geolocation method Backstrom et al. [2010] and with choosing the most frequent location from within the friends ('Most Common Location') or weighted by number of users in each city ('Inverse City Frequency'). Using a SVM classifier with a Radial Basis Function (RBF) kernel we obtain the best results, while adding more features improves results.

Method	≤ 0 **m**	≤ 50 **m**
Random assignment	$31.6\% \pm 0.4\%$	$43.6\% \pm 0.5\%$
Backstrom Backstrom et al. [2010]	9.1%	47.3%
Backstrom Best Backstrom et al. [2010]	25.4%	52.0%
Most Common Location	$39.4\% \pm 0.3\%$	$47.7\% \pm 0.4\%$
Inverse City Frequency	$10.6\% \pm 0.1\%$	$40.5\% \pm 0.4\%$
SVM: Population	$45.9\% \pm 0.4\%$	$58.2\% \pm 0.4\%$
SVM: Population & Triads	$47.1\% \pm 0.4\%$	$59.2\% \pm 0.4\%$
SVM: Population & Triads & Reciprocation	$\mathbf{50.0\% \pm 0.5\%}$	$\mathbf{62.0\% \pm 0.5\%}$
Oracle performance	$78.1\% \pm 0.2\%$	$89.5\% \pm 0.2\%$

Table 3.5: Performance of multiple geolocation methods, with 99% confidence intervals.

The results show significant improvement to the previous method using the social network connections and, combined with the other two precision focused methods, we can achieve geolocation performance comparable with geolocation using the IP address, which is inaccessible to regular users.

3.1.8 Possible extensions

In addition to the modes previously described that are already an integral part of the general framework for data pre-processing, other tools have been identified as useful for other applications. Here, we list these modes as well as viable existing tools that can be readily integrated.

Filters Filtering of messages restricts the input dataset to a subset that respects a specific set of conditions. These are usually in the form of existing fields having a value from a set or belonging to an interval. For example, existing filters can restrict a dataset to contain only data detected as belonging to a specific language. This can be extended to contain a specific a set of tokens, to answer a query or to be opinionated.

POS Tagger This is a tool that labels each word in a text with its appropriate part-of-speech (e.g. noun, personal pronoun). Studies have identified [Gimpel et al., 2011, Derczynski et al., 2013a] that state-of-the-art POS taggers suffer a drop in accuracy when applied on OSN data. Tools adapted to OSN data have been developed by Gimpel et al. [2011], Owoputi et al. [2013], Derczynski et al. [2013b] and are freely available.

Named entity recogniser The NER is a tool that extracts the tokens that belong to categories like names of persons, organisations, locations, etc. Ritter et al. [2011] reveals that the results of standard NER systems on OSN data have low accuracy. It also builds an improved freely available tool for this task. Also, systems participating in the MSM 2013 Concept Extraction Challenge [Cano et al., 2013] represent relevant alternatives for solving this task.

Normaliser Text normalisation refers to the procedure which assigns the standard vocabulary form to a non-standard lexical variant. The later are very present in OSN data and this is usually a consequence of the need for conveying a more nuanced opinion (e.g. word lengthening for amplifying opinions [Brody and Diakopoulos, 2011]), regional or demographic peculiarities (e.g. 'sis' is usually a term specific to African American people [Eisenstein et al., 2011]) or the length constraints of the system (e.g. 4 instead of 'for'). Example of normalised versions of texts are presented in Table 3.6 The normalisation route is appealing because it helps alleviating word level sparsity and should theoretically lead to a reduced effort in adaptation of other processing

tools. However, some researchers [Eisenstein, 2013] have questioned the need of normalisation or have used features that deal with un-normalised text [Owoputi et al., 2013].

There are multiple approaches to text normalisation [Han and Baldwin, 2011, Han et al., 2013, Ling et al., 2013, Yang and Eisenstein, 2013, Xu et al., 2013], exploiting different aspects like pronunciation, edit distance, translations into another language. Han et al. [2012] showed that a simple dictionary based approach can achieve a reasonably high performance. Any of these systems can be added as a mode that outputs a list of normalised variants of the tokens.

Original	RT @USER: Ah!Had d most amazing meal,Love WASABI at th Taj,defntly d best japanese food in India:)had 2 loosen my belt ate so ...
Normalised	RT @USER: Ah! Had the most amazing meal,Love WASABI at the Taj,definitely the best japanese food in India:)had to loosen my belt ate so ...
Original	@erinbieberxoxo ok:)i'll keep thinking.wat did i say b4?haha.i 4got.
Normalised	@erinbieberxoxo ok:)i'll keep thinking.what did i say before?haha.i forgot.

Table 3.6: Normalisation examples.

User influence All the OSN peculiarities that make it popular for sharing timely updates, like its open structure, ease of usage and access, raise natural issues about user credibility [Castillo et al., 2011] and content attractiveness [Suh et al., 2010]. User influence or impact is thus an essential step in order to asses the importance of a given message in social media. This can be established in a number of ways. Intuitively, the raw number of users that receive a message (i.e. the 'followers') is a natural candidate. But, in the case of Twitter, this might not be the case, because tweets can for example be retweeted, reaching a much bigger audience. Meeyoung et al. [2010] compare followers, retweets and @-mentions received by an account as measures of influence and discover that the number of followers alone is not always a good indicator of influence. Furthermore, even a regular user that does not post content can gain a great number of followers using, for example, a follow-back strategy (i.e. an account follows a user expecting reciprocation from the other). Bakshy et al. [2011] measures influence of an account as the logarithm of the average size of all cascades for which that user was a seed. Companies like PeerIndex[4] or Klout[5] also offer influence scores for users which can be accessed using their public APIs.

[4] `http://www.peerindex.com/`

[5] `http://www.klout.com/`

In [Lampos et al., 2014] we propose a new impact score which is shown to relate to popular perception. Impact is usually quantified by the number of followers, i.e. users interested in this account. However, non popular users can also gain a large number of followers, for example by exploiting the follow-back strategy, i.e. following many other accounts expecting to get followed back. The ratio of followers and followees is not a reliable metric, as it is invariant to scaling. In order to solve this, the number of followers is emphasised by squaring. An additional impact indicator is the number of times an account has been listed by others. Lists provide a way to curate content, thus, users included in many lists are attractors of interest. The final impact score is defined as: $\ln\left((listed+1)\,(followers+1)^2/(followees+1)\right)$. Further, we propose a learning method for predicting and characterising impact of users based on features under a user's direct control (e.g. topics of tweets, user profile features, not the number of retweets) obtaining as high as $\rho = 0.78$ correlation with the impact score. Bakshy et al. [2011] achieve a modest correlation ($\rho = 0.34$) with their impact score (based on number of initiated cascades of posts) using classification and regression trees with a restricted set of features and past user influence. Any of these impact scores can be added as a mode and further used to filter documents based on a minimum influence threshold.

3.2 Open-source tool

This section describes the technical details of our OSN data processing architecture and presents the main aims for which it was built. Our system is focused on a real world scenario where fast processing and accuracy is paramount, while also allowing users to easily add their own modules and specify which processing steps they want when running this system. We identify two main use cases. Firstly, batch analysis of large volumes of documents and applying filters for keywords, language, etc in order to compute aggregate counts of features, sentiment, etc over them when dealing with archival scenarios. Secondly, a real time analysis scenario with streaming input. To address these concerns, we propose a set of command line tools. The tools implement the modes of our pre-processing pipeline presented in Section 3.1.

3.2.1 Aims

The aims for our architecture for OSN data pre-processing are as follows:

Scalability The volume of data we need to process is expected to be very large in size and analysis might take less time than random disk access;

Modularity Each data analysis step should be a separate tool. The framework will be able to combine separate tools (modes) and allow easy extension with new modes;

Pipeline The end-user should have control over what processing steps are required for their application. The system should have a pipeline architecture with interchangeable modules;

Compatibility The system should be able to work with data from different sources (e.g. Twitter, Google+), which may have different format and data fields;

Data consistency In order for different modes to be able to run independently or part of a chain, the original data fields should never be altered;

Re-usability To guarantee the repeatability of all our experiments, the architecture should be released as open-source.

3.2.2 Architecture

We expect that any specific task applied to an individual tweet will take less time compared to random disk access required for reading and outputting compressed data. This I/O bound analysis has been addressed in the past[6] with the use of clusters of machines with shared access to distributed data using the MapReduce framework and Distributed File System. MapReduce is a software framework for distributed computing introduced by Google in 2004 to support distributed processing of massive datasets on commodity server clusters. Logically, the MapReduce computational model consists of two steps, Map and Reduce. In the Map step, data items are mapped in parallel to $< key, value >$ pairs, which are then aggregated, sorted by key and sent as input to the Reduce step which performs and aggregation on all data and also outputs $< key, value >$ pairs.

We use Apache Hadoop,[7] an open-source implementation of MapReduce. Specifying the Map and Reduce functions for Hadoop can be done using either Hadoop Streaming[8] or by writing custom Java tools interacting with the underlying Hadoop

[6] `http://engineering.twitter.com/2010/04/hadoop-at-twitter.html`
[7] `http://hadoop.apache.org/`
[8] `http://hadoop.apache.org/common/docs/r0.15.2/streaming.html`

Java libraries. Hadoop streaming allows the specification of the Map and Reduce functionality through POSIX like standard in and standard out enabled command line utilities. This allows for quick prototyping using any programming language, but lacks the flexibility exposed when using Hadoop as a library in Java. Instead we implement a Hadoop enabled pre-processing tool in Java. This tool exposes the various stages of our pre-processing pipeline as modes. The inner components of the tool are shared between two separate tools: a local command line utility and a Hadoop processing utility. The individual stages of the pre-processing pipeline are implemented in pure Java and exposed as modes in the tools. In the local utility, individual tweets are loaded one at a time and each selected pipeline stage is applied to a tweet as it is loaded. In the Hadoop implementation, the Map stage is used to load each item wherein each pre-processing step is applied to individually and emitted by the Mapper. If an aggregation is required, this is performed in the Reducer, otherwise the Null Reducer is used.

For addressing the modularity aim, both the Hadoop and Local tools are driven through the same mode specifications and implementations. To implement a new mode in both tools, a simple Java interface is implemented that specifies a single function which accepts a data item and adds its output. Furthermore, multiple implemented modes can be executed in a single invocation of the tool. Concretely, this results in multiple analysis being performed on a single item while it is in memory (in the Local tool) or multiple analysis being performed in the single map task (in the Hadoop tool).

3.2.3 Data format

Data consistency is related to modularity. Each component of the pipeline may run in isolation, or as a chain of pre-processing tasks. To this end, each component must be able to predict what data is available and be able to reuse or reproduce the output of preceding stages it relies upon. This means that the original data item must remain unchanged. To achieve this, we keep the format of the original data and at each step of the pipeline we augment the output of the previous steps with extra fields inside a special 'analysis' entry corresponding to the results of preceding steps. Implemented components of the pipeline use this 'analysis' map to retrieve the output of previous stages[9] and they also add their own analysis output to this map.

[9]A stage has a unique name which it uses to store data.

The system was first built on using Twitter data and the JSON data format provided by the majority of OSNs as output from their API. However, although there are similarities between the type of information provided by each OSN, each might have a different format schema (e.g. different data fields or different names for the same field). In order to address the compatibility aim, we internally use the USMF (Unified Social Media Format) data format. USMF is a data format that generalises data extracted from social networks devised by Tawlk.[10] A specific OSN data schema can be converted to this universal format for social media items by defining a simple mapping between fields.

3.2.4 Running times

The results in Table 3.7 show timings of both our Standard and Hadoop pre-processing tools. Both experiments were run on tweets generated in one day on October 10th, 2010. The local tool was run on a single core whilst the Hadoop tool was run on a Hadoop cluster of 6 machines, totalling 84 virtual cores across 42 physical cores. Our timings show that a relatively small Hadoop cluster can pre-process tweets at a faster rate then what is created in the live Twitter stream. Furthermore, due to Hadoop's ability to scale with the addition of new machines, we believe that the addition of a few machines will allow our tools to scale easily as Twitter grows in popularity.

Time	**Local**	**Hadoop**	**10% Twitter**	**Total Twitter**
1 hour	0.51	7.6	∼1	∼10

Table 3.7: Number of tweets (in millions) analysed and created in an hour. Analysis performed: tokenisation and language detection.

For re-usability, we make out pre-processing pipeline available to the community for use in reproducing our experiments or producing novel applications. This is done via an open-source project under BSD licence.[11]

3.3 Conclusions

We have presented in this chapter a framework that includes multiple modes for processing OSN data. These modes include standard NLP operations like tokenisation and language detection as well as others based on learning over the social network structure as in the case of user geolocation. The framework that integrates all the

[10] `http://sharismlab.com/blog/2012/11/16/usmf-one-format-to-rule-them-all/`
[11] `http://github.com/danielpreotiuc/trendminer-java`

modes was developed as an open-source project and can reliably process large volumes of streaming data in parallel in a pipeline architecture.

By applying all the processing steps we have access to a richer and standardised format of OSN data. This representation is suitable for filtering, grouping and to build feature representations that can be fed to applications downstream.

In the next chapters we will use the collected and processed data as input for algorithms that explicitly model the temporal dimension. First, in the following Chapter 4, we analyse data temporality in a supervised setting by identifying correlations to real-world indicators.

Chapter 4

Temporality and social media

OSN data is inherently temporal with timestamped posts arriving in a stream, as we have seen from previous chapters. But, what is the impact of temporality and how should we model its effect? The temporal characteristics of data were previously studied by researchers in different settings e.g. by simply examining timestaped data, by using it to forecast values internal or external to social media, by incorporating time into graphical models.

Timestamped data can be analysed without explicitly modelling time, but only analysing its effects. An overview of these approaches is performed in Section 4.1. Basic methods presented in Section 4.1.1 aim to analyse existing temporal patterns by examining past data. Often an application on top of this type of analysis is forecasting of variables in the near future (e.g. the following day) based on OSN derived features. These applications are presented in Section 4.1.2. In a supervised setting, a model is learnt on existing previous data and is used to predict the output variable when its values are not available. In the case of OSNs, the output variables can be internal (e.g. if a keyword will trend, frequency of people visiting bars tomorrow) or external, with which some of the system's features are expected to correlate (e.g. rainfall, natural disasters). More sophisticated models aim to model time as a separate variable (e.g. conditioning on time or including the timestamp as a variable in a generative model) before analysing its dynamics. However, modelling time is usually performed by making strong assumptions, such as temporal smoothness. In Section 4.2 we review work on temporal modelling, with an emphasis on OSN oriented research.

In Section 4.3 we explore how text features derived from Twitter data can be correlated to real world indicators external to the social network. We consider time varying political party voting intention scores from two different countries as use cases. In a supervised setting, we aim to forecast these scores from Twitter data. We build sparse regression models that learn which features are predictive of the underlying

signal. In order to handle the large amount of input noise we introduce bilinear regression models in Section 4.3.3. However, these are not temporal models as they treat the word features as stationary. In Section 4.4 we explore one of the simplest types of temporal dynamics: an online setup where old data is down-weighted in favour of new data. This way, we aim to show that temporality plays an important role in building predictive models from social media.

4.1 Non-temporal models

4.1.1 Timestamped data analysis

A straightforward method of analysing the temporal characteristics of features derived from OSN data is to perform time series analysis. Tools that enable this kind of analysis are available for large scale data, for example in the case of search queries and news headlines[1] or n-grams in book collections.[2]

Visualisation Further work on data visualisation in OSNs have shown informative patterns of human activity. Rios and Lin [2012, 2013] present tweeting volume patterns over a year in different cities of the world. They display low activity during the nights and usually the highest activity during the evenings. Weekends exhibit a different pattern compared to weekdays, with the activity more evenly spread out during the course of the day. However, cultural peculiarities are also a factor that influences these patterns. For example, denizens of some cities (e.g. Tokyo) are more 'serious' and do not tweet that much during usual work hours (10AM - 6PM) while school holidays mean that more tweets are sent later in the night as a large group of people are not constrained to wake up early in the morning. Religious customs are shown to have an impact as well, as in cities with a majority of Islamic population, people do not tweet much during prayer times or adjust their daily program during Ramadan to have more activities after sunset. When looking at human activity from LBSNs in Section 2.2.3, we have also identified similar patterns for the weekdays and weekends, while very different across these categories. However, other than geo-tags and time, LBSNs offer other meta-data about the location of the posts, such as the type of place (e.g. bar, park). A joint analysis will be presented in Section 5.3.

[1] `http://www.google.co.uk/trends/`
[2] `http://books.google.com/ngrams/`

For OSNs, multiple commercial tools have been developed to enable users to discover and analyse trends.[3,4,5,6] Usually, these include a filtering step which restricts the content to that relevant to the users interests. Then, aggregate statistics such as volume of messages, relevant terms and sentiment scores are displayed as a time series for human interpretation.

Filtering Some research studies use the same filtering approach for analysis. Rios and Lin [2012] display activity during a major event, the Football World Cup in 2010, highlighting the volume of activity for different teams and competition stages. By monitoring keywords over time, Doan et al. [2011] study the relationship between Twitter messages and the Tohoku Earthquake in Japan and find that Japanese messages could be used as an early warning system. They also measure anxiety in the Japanese population using keywords and found that a spike occurs in the case of a natural disaster, but this goes back to normal levels after a short time span. Lansdall-Welfare et al. [2012] filter geo-located tweets in the U.K. for mood-bearing words (four moods: joy, fear, anger and sadness [Strapparava and Valitutti, 2004]) to identify the 'mood of the nation'.[7] They show that bursts in certain moods coincide with events, such as the U.K. riots or Christmas. Dodds et al. [2011] attempt to measure happiness as self-reported by users on Twitter. With the help of Amazon Mechanical Turk[8] they have gathered happiness evaluations for more than 10,000 words and used these to analyse specific words and create an aggregate 'happiness' profile for different time intervals. For example, they have shown an increase in happiness during mornings and in the weekends, confirming also conclusions of [Golder and Macy, 2011]. The changes observed in moods over time lead them to perform a word shift analysis. This looks at which words contribute to the change in average happiness for days in which a sudden change in this mood was observed. For example, words like 'kiss', 'prince' and 'dress' contributed the most for the happiness spike on the day of the British Royal Wedding of 2011.

Event detection A similar idea to word shift analysis was used for event detection in [Mathioudakis and Koudas, 2010]. They assume that if a word exhibits a burst in usage, then an event relating to this word has occurred. Snowsill et al. [2010]

[3] `http://www.brandwatch.com/`
[4] `http://www.opiniontracker.net/`
[5] `http://www.ubervu.com/`
[6] `http://www.tribatics.com/`
[7] `http://geopatterns.enm.bris.ac.uk/mood/`
[8] `http://www.mturk.com/`

perform a statistical test to identify which n-grams have increased in frequency over their expected frequency, thus indicating possible underlying events. The different temporal patterns of social media memes have been explored by Yang and Leskovec [2011]. They build a K-Spectral Centroid method for grouping time series which uncover discovering six categories, all of which have a large burst as a component. Models that predict the volume of the temporal diffusion of information in OSNs have also been studied [Yang and Leskovec, 2010] showing that imitation and novelty are key factors in the adoption of news-related textual words and phrases.

Dimensionality reduction The major issue with analysing trends of individual words is their large number. A better representation of text content is arguably needed in order to enable a better overview of the OSN data stream [Ramage et al., 2010]. A traditional approach is to distill a collection of documents into clusters of words or 'topics'.

Topic models are generative probabilistic models that learn a small number of distributions over a words (called 'topics') which describe a document collection. Latent Dirichlet Allocation (LDA) [Blei et al., 2003], the most popular topic model, is a fully unsupervised model that uses co-occurrences between words in a document in order to group them in similar 'topics'. A document is then represented by a mixture of these 'topics'. Because each document can belong with a certain probability to each underlying topic, topic models have been used by researchers to aid browsing document collections.[9,10,11]

Thus, it is not surprising that they have also been used to analyse collections of timestamped documents. Griffiths and Steyvers [2004] have trained LDA over a large set of PNAS scientific articles and plot the distribution of topics over time. Hall et al. [2008] try to link topic distributions between documents from different years in order to analyse trends in the NLP community. However, all these models only use the timestamp after learning the distributions as a way of displaying the information, rather than incorporating it as a variable of the learned model. An overview of work that uses the timestamp in learning is presented in Section 4.2.4.

Use of topics models in OSNs has been restricted mostly due to the large data volume and the restricted length of a document, which impairs inference due to a low number of co-occurring terms. Hong and Davison [2010] apply LDA in order to model OSN user interests and to aid categorisation. They have also shown that

[9] `http://www.cs.cmu.edu/~lemur/science/`
[10] `http://www.princeton.edu/~achaney/tmve/wiki100k/browse/topic-presence.html`
[11] `http://christo.cs.umass.edu/NYT/`

another model, the Author-Topic model [Rosen-Zvi et al., 2004], quite successful for identifying authored sections in multi-authored documents, is useless when confronted with OSN data. Zhao et al. [2011] introduce TwitterLDA is based on the assumption that one tweet can only have words from one topic and a 'background' one. This shows better results than when applying standard LDA. They compare the topics learnt with LDA over newswire with those from social media, showing a completely different distribution of topics between the two sources, with personal and family related topics more prominent on Twitter. An usual method of overcoming the short length of OSN messages is to aggregate them based on a similar characteristic, such as the author [Eisenstein et al., 2010], and use them as documents in regular LDA.

4.1.2 Forecasting

As we have seen in the previous section, many studies provide evidence that meaningful patterns can be extracted from the data of the OSN stream. These patterns can be harnessed in forecasting settings. A model is learnt using data coupled with responses and used to forecast the response in the future, based only on data. Although these do not represent temporal models per se, they use timestamped data in order to predict future behaviour, thus using the temporality of the data.

For example, a very popular application is that of predicting future human behaviour based on the mobility history, also known as 'next location prediction' [Chin and Zhang, 2013]. Methods used in mobile computing consider the spatial trajectories [Spaccapietra et al., 2008], temporal patterns [Nguyen and Tu Minh, 2007], or spatio-temporal patterns [Scellato et al., 2011a] for location prediction. For LBSNs, social proprieties (e.g. friend's patterns, demographic data) have also been incorporated into prediction models based on spatio-temporal patterns in [Chang and Sun, 2011, Gao et al., 2012] with relative success. For Twitter, researchers have attempted to predict future events, such as the volume of hashtags using content features [Tsur and Rappoport, 2012] or if a message is going to be retweeted [Petrović et al., 2011] or deleted [Petrović et al., 2013].

In the previous section we have also shown that researchers used OSN data in order to identify and observe real world events. We can thus expect that response variables which are external to the OSN can be nowcasted or forecasted using OSN derived features. Nowcasting or 'predicting the present' [Choi and Varian, 2011] is used to track current indicators in near-real time for responses that are usually very costly to evaluate and can further offer a detailed break-down with the help of metadata (e.g. geography, demographics). Forecasting or 'predicting the future' is use to

predict variables days or weeks in advance. Usually the response variables are from the domain of Social Sciences [Miller, 2011] such as political science (e.g. voting intention [Tumasjan et al., 2010], presidential approval [O'Connor et al., 2010]), economics (e.g. stock market trends [Bollen et al., 2011]) or health (e.g. flu rates [Lampos et al., 2010]).

Before social media, Goel et al. [2010] have used search query volumes on Google to forecast the opening weekend box-office revenue for features films, songs, video games, while Ginsberg et al. [2008] used query volumes to track flu trends. By using OSN data, successful attempts have at first been reported for a range of applications with easy to derive features. Asur and Huberman [2010] suggest that the tweet rate about a feature film can predict the box office revenues better than traditional forecasting models. Tumasjan et al. [2010] show experiments on the German federal election of 2009 and find significant correlation between the number of political party mentions on Twitter and election results. Culotta [2010] shows strong correlation between flu related tweet volume and CDC ILI (The U.S. Centers for Disease Control and Prevention infuenza-like illness) rates.

Methods building on the previous perform an analysis on the filtered set of messages. The most common is to compute a sentiment score based on the intuition that more favourable messages toward some entity will correlate better with the response variable. For financial time series, Bollen et al. [2011] suggested that Twitter can be used for stock market prediction. With political voting intention application, O'Connor et al. [2010] and Lampos [2012] filter the Twitter stream and compute a sentiment score (using OpinionFinder [Wilson et al., 2005] or WordNet Affect [Strapparava and Valitutti, 2004]) for each political entity (party or politician). They obtain mixed results, with the OSN derived features not correlating in all cases with the response variable, hinting that more complex methods may be needed.

In fact, most of the initial research was subject to controversy due to the lack in experiment replicability, one-sample testing or not simulating a pure forecasting setup by feature engineering using the test results [Metaxas et al., 2011]. For example, Jungherr et al. [2012] contradict the conclusions of Tumasjan et al. [2010] by showing that the volume of tweets about the Pirate Party was the largest across all parties. Had they been included in the original study, they would have been forecasted as the winners of the election although they only obtained 2% of the votes. Metaxas et al. [2011] perform an analysis over 6 different U.S. senatorial or congressional races using the methods of [Tumasjan et al., 2010] and [O'Connor et al., 2010]. They conclude that any of the two methods predicts only half of the elections correctly. By

comparing the predictions, the sentiment based methods of [O'Connor et al., 2010] performs consistently better, but still has an average error of 7,6% percentage points to the actual voting intention (compared to 17,1%).

A further limitation is the dependence of some methods on large hand crafted lists of words for filtering or sentiment calculation. Most of the systems are thus not easily portable across different tasks and data scenarios. In order to address this and also to obtain a better accuracy, Marchetti-Bowick and Chambers [2012] use distant supervision [Mintz et al., 2009] for both the filtering and sentiment steps. This involves using a few general keywords in order to retrieve relevant documents which are then used to train a classifier in order to extract further messages that even if they do not include the original keywords. For presidential approval in the U.S., they discover that distant supervision aids both steps, obtaining a high positive correlation ($\rho = 0.70$) with both positive and negative approval rate compared to the method of [O'Connor et al., 2010] which shows negative correlation with positive approval. However, they also discover that computing sentiment over the entire data, without content filtering, is comparable to their best results.

Several issues have been identified in existing forecasting models [Gayo-Avello, 2012]. The problem of demographic and self-selection bias in the OSN user base should be incorporated as a factor and predictions should be sensitive to this. Statistics have shown that OSN population is not representative of the entire population[12] and reactions to events are often disproportionate to the public.[13] In 2011, only 22% of registered voters have let others know how they voted on OSNs in the U.S.[14] Further issues are represented by the vast volume of irrelevant messages and the difficulty in identifying relevant messages amongst these. Opinion spam and propaganda are ever increasing phenomena which should also be accounted for [Mustafaraj et al., 2011]. Models should compare their performance to traditional methods and the processing operations on the data before the prediction should be pre-determined and fixed.

A few methods try to address some of these issues. Huberty [2013] used an ensemble supervised learning algorithm on OSN data for the 2010 U.S. congressional elections to obtain better predictions then a very competitive incumbency baseline and the volume and sentiment based methods. However, testing the same model learnt on out-of-sample 2010 data on 2012 election data showed poor results. The conclusion is that recent data is important in forecasting. In a similar regression setup, Lampos

[12] http://pewinternet.org/Reports/2013/social-networking-sites.aspx

[13] http://www.pewresearch.org/2013/03/04/twitter-reaction-to-events-often-at-odds-with-overall-public-opinion/

[14] http://pewinternet.org/Reports/2011/Politics-and-social-media.aspx

and Cristianini [2010] explore the relationship between flu rates as measured in the U.K. and data from Twitter. They use a regularised linear regression [Bach, 2008] in order to identify a sparse set of keywords that best correlate in frequency with the flu rates. They achieve as high as $\rho = 0.94$ correlation with ground truth in a forecasting setting [Lampos et al., 2010]. We argue that by using existing ground truth data we can automatically adjust for the demographic and self-selection biases and perform automatic word features selection, removing the need for adaptation between test scenarios. We explore a model based on this idea in Section 4.3.3. Further systematic analysis of forecasting models are presented in [Schoen et al., 2013, Gayo-Avello, 2012].

4.2 Temporal models

4.2.1 Online learning

Temporal models aim to explicitly incorporate time as a factor that impacts the outcome of the model. This is based on the concept of data non-stationarity: the properties of the data are different conditioned on time. Central to this are methods and concepts of online learning.

In online learning, data arrives in a stream and needs to be incorporated into the model one at a time. Usually labeled feedback is also provided with data in a streaming setting. This is in contrast with the traditional machine learning setup where the data is static and a model is trained on the entire dataset. Models trained in this way are not able to incorporate new information without retraining the entire model. The most naive way in extending these models to handle streaming data is to retrain models in fixed size data windows. However, this method is usually hard to apply in real settings due to the cost in time associated with training a new model from scratch. This motivates research into online one pass algorithms trained over streaming input.

In the general online learning framework, no assumptions on the origin of the data stream (e.g. stationarity) are made. Studies have shown that in practice, dataset size is an important factor for the performance of different models [Halevy et al., 2009]. Simpler models trained on large data have produced better results than better models with a smaller dataset [Banko and Brill, 2001]. Thus, evaluation of online learning methods is usually done by *regret analysis* [Shalev-Shwartz, 2007]. Regret analysis compares the cumulative loss of the sequentially trained online learning algorithm to the results of a batch model trained from scratch from all the data available before

each prediction. Regret thus measures the penalty paid by the online model compared to the batch model. An alternative is to consider that the model should evolve over time and the online algorithm is penalised in both loss and the shifts in the model [Herbster and Warmuth, 2001]. Negative regret is usually hard to obtain in static datasets, where one pass online algorithms are expected to perform worse than the batch variants. However, the potential non-stationarity of the data might lead to better results by an online learner.

Recently, online variants of learning algorithms have been proposed. Stochastic Gradient Descent is a common approach to online learning that has shown very good results in NLP tasks [Bottou, 2010, Martins, 2012]. For graphical models, online variational Bayes was used to build an online version of LDA which finds models as least as good as those learned with batch variational Bayes [Hoffman et al., 2010]. An alternative approach to an online setup for LDA is presented in [AlSumait et al., 2008] and a version that differs from the original version by considering a sliding window for training is presented in [AlSumait et al., 2009]. A paper that adapts pLSA [Hofmann, 1999], a basic version of LDA, to an online setting is presented by Gohr et al. [2009]. Applications using OSN data are relatively restricted, as many researchers have only tested batch prototypes.

4.2.2 Online clustering and event detection

Topic (or event) detection and tracking has experienced a new rise in attention once with the availability of OSN data. Historically, these tasks were defined and studied in the context of 'Topic detection and tracking' (TDT) on newswire text [Allan, 2002]. Methods for topic detection can be grouped into document and feature based approaches. The former aims to cluster documents into events and then extract features from the clusters [Brants et al., 2003]. The later category is based on identifying and clustering features that are representative of events [Kleinberg, 2002]. In OSNs, a streaming setup is assumed as data arrives ordered in time and in high volume, making one pass algorithms a necessity.

Generally, clustering methods that operate in a streaming setting have been adapted from batch algorithms [Guha et al., 2003]. The two main approaches are centroid models and density based models. Centroid models assume that each cluster is represented by a central item. The most common method of this type fixes the number of clusters beforehand to k, leading to the k-median optimisation problem, usually solved by the k-means algorithm. Online approaches to this problem have

shown very good results [O'Callaghan et al., 2002]. However, the restriction of specifying k beforehand is limiting for discovering new topics, usually meaning that in order for a new cluster to emerge, another has to be deleted. Another limitation is usually the spherical shape of the clusters which is due to the distance metric (i.e. usually Euclidian).

Density based models (e.g. DBSCAN [Ester et al., 1996]) assume that clusters are represented by areas of high density. In order to compute the 'density', a notion of similarity between two items is used and needs to be defined. Items that do not belong to any high density area are considered noise and thus not clustered. Density based methods have the advantage of handling outliers and can identify arbitrary numbers and shapes of clusters. However, the algorithms are based on a good definition of similarity, which might not be intuitive for high-dimensionality problems and are incapable of identifying clusters of different densities. Streaming versions of density based algorithms have also been proposed [Feng et al., 2006]. Evolutionary clustering methods [Chakrabarti et al., 2006] provide another view on temporal clustering. They attempt to link clustering results performed on consecutive time intervals. Results aim to reflect long-term trends while being robust to short-term variations. This is performed by adding a temporal smoothness penalty to the static clustering criterion in each time interval [Chi et al., 2007, Xu et al., 2010].

Document based methods for online OSN data clustering usually use the key assumption of temporal locality due to the ephemeral nature of most events, as was identified in Section 4.1.1. This is used by creating time windows for clustering in order to allow a quick change in topics and to be affective when presented with large data volumes. For example, Incremental DBSCAN [Ester et al., 1998] is used by [Lee, 2012] for clustering within a sliding window, moreover adding a penalty for items that are further apart in time. For a related task, that of first story detection, [Petrović et al., 2010] uses randomized algorithms (i.e. Locality Sensitive hashing) to identify messages in OSNs that are very different from all the previous and use hashing to keep track of documents.

For feature based methods, a parametric distribution of the temporal patterns of events is usually assumed e.g. a burst in frequency followed by a decay. By analysing natural hazard related keywords, Sakaki et al. [2010] assume that the distribution of messages relating to these events has an exponential distribution. They build a classifier for these messages and build an early detection system for this type of events. [Becker et al., 2011b] assigns a message to a cluster based on document similarity. However, in order to discover which clusters are related to events, they use

temporal features such as the similarity of a word temporal profile to an exponential distribution. Mathioudakis and Koudas [2010] identifies bursts in frequencies of words and assumes these are caused by events. The method further filters the stream for the bursty words and finds other words which co-occur with it in order to characterise an event by a set of words. Zimmermann et al. [2012] tries to address limitation in findings subtopics by creating a two-level hierarchy of topics and identifying whether a change occurred either at a local or global level. Li et al. [2012] expand burst detection to segments in messages and identify the top segments as potential event-related segments. Clustering using a variant of k-nearest neighbours is used in order to group the features and further refine the clusters.

Feature based methods are usually easier to scale and clustering in a high dimensional space with very few activated features (non-zero values in the vector representation) usually leads to weak cluster assignments. However, the assumption of a bursty behaviour for feature frequencies might not always hold, for example in the case of frequent features or topics without a large coverage where bursts are hard to be reliably identified. In Chapter 6 we attempt to overcome these limitations by identifying events based on variation in co-occurrences. We argue that these are robust to word frequency values and can also capture events of restricted interest.

4.2.3 Regularised learning

One of the common approaches to regression and classification is using generalised linear models [McCullagh and Nelder, 1989]. Models take a m-dimensional feature vector input $\boldsymbol{x}$ and aim to build linear functions of $\boldsymbol{x}$ for predicting the response variable y. The objective is to find an optimal function (in the form of a vector of weights $\boldsymbol{w}$) that minimises a model-dependent loss function L subject to a regularisation penalty ψ. The regularisation penalty incorporates into the model desirable properties of the weight vector $\boldsymbol{w}$. For example, by using a l_2 regulariser, high weights are penalised (a.k.a. 'ridge regression' [Hoerl and Kennard, 1970]). This is very important for high dimensional problems in order to avoid overfitting. Another option is to use the l_1 regulariser (a.k.a. LASSO [Tibshirani, 1994]), which penalises more harshly non-zero values for weights. This builds into the model the bias that only a sparse set of features have to be used for prediction. This is desirable when m is very large, usually higher than the number of training examples n and several features are expected to be correlated.

However, the regulariser can also be used to incorporate time dependencies between features. Assuming that features can be arranged in temporal order, fused

LASSO [Tibshirani et al., 2005] penalises the l_1 norm of the weights and their consecutive differences. The l_1 penalty models local smoothness between weights. On the assumption that the training examples can be grouped based on time, [Yogatama et al., 2011] learn a weight vector for each time step. They use l_2 regularisation between the weight vectors in each separate time step to induce smoothness for weights over time. They note that the prior distribution over the weight sequences is linked to a first-order auto-regressive process, AR(1).

4.2.4 Generative models

Modeling time using generative models, such as LDA, has been studied in different contexts. We have identified two major ways in which researchers have addressed the issue of time modeling: by expressing the time variable explicitly in the model or by building into the model the time dependency between variables.

The former category of models that we have identified consists of models that explicitly treat the time variable, including it into the graphical models specification. The first model that used this was Topics Over Time (TOT) model introduced in [Wang and McCallum, 2006] where a beta prior is assumed over the temporal distribution. The model presented in [Li et al., 2006] uses the same idea and priors as in the Topics over Time model. However, they use as the underlying model, instead of LDA, Pachinko Allocation [Li and McCallum, 2006]. Song et al. [2005] suggests a topic model that incorporates different variables in the model (e.g. contact information, content and the time of a text). Use cases include timestamped e-mail collections, the set of the State of the Union addresses or a set of conference proceedings.

The later category is characterised by the intention of building into the model a dependency between documents or collections of documents. For many collections of texts the implicit assumption of exchangeable documents, which is an integral part of LDA, is inappropriate as is the case of collections that come from different time periods. The first study to consider this problem is Blei and Lafferty [2006]. The goal is to capture the evolution of topics in a sequentially organised corpus of documents. The paper introduces Dynamical Topic Models (DTM) where documents in each time bucket are modeled with LDA and the topics associated with the current bucket evolve from the topics associated with the previous one. Building upon DTMs (or discrete dDTM), other models like cDTM (Continuous Time Dynamic Topic Model) [Wang et al., 2008] and Multiscale Dynamic Topic Models (MDTM) and Online MDTM [Iwata et al., 2010] are introduced. Online pattern discovery in multiple time series

is the goal of Wei et al. [2007]. This study presents the Dynamic Mixture Models (DMM), a latent variable model that takes into consideration the timestamps of data records in dynamic streams.

Markov topic models (MTM) are introduced in [Wang et al., 2009]. These models aim to learn topics simultaneously from multiple corpora. MTMs are an extension of the LDA where a Markovian framework is applied to the topic parameters for different corpora. Using the exact same framework, but only modeling topic distributions with Markov random fields, are the Markov Random topic fields (MRTF) introduced by Daume III [2009]. While not modeling temporal data in their experiments, these two models (MRM and MRTF) can be adapted to temporal modeling by assuming the dependencies of documents or corpora to be of temporal origin. These studies use datasets of long and well structured documents on a restricted set of topics (e.g. conference proceedings or political addresses).

4.3 Forecasting political voting intention

In this section, we approach the problem of supervised learning with the goal of forecasting external variables based on textual features derived from OSNs. We develop models that go beyond state-of-the-art techniques. Specifically, the bilinear modeling framework is introduced for handling the noisy nature of OSN data, where only a small fraction of the content is relevant for the predictive task. As case studies, we consider the application of predicting political party voting intention in two different countries: the U.K. and Austria. We compare our models to existing methods, such as volume, sentiment score and regularised linear regression. These are expected to suffer in terms of predictive accuracy when faced with data of large scale and in presence of noise.

4.3.1 Voting intention scores

We use as the external variable for prediction voting intention percentages for the top political parties. In order to prove that our methods are generalisable and robust, we use two different test cases: the U.K. and Austria. These are bound by different characteristics such as language, time intervals, number of users, size and relevance of data. We use as data the U.K. Users Dataset and Austrian Politics Dataset introduced in Section 2.1.3.

Voting intention is quantified by polls performed by specialist polling institutes in both countries. First, for the U.K., we use polls provided by YouGov[15] which usually provide 5 polls every week, totaling 240 polls for the almost 2 years of the dataset. We selected the top 3 parties, which accounted for more than 90% of the voting intention in our interval of study: the Conservative Party (CON), the Labour Party (LAB) and the Liberal Democrats (LIB). For Austria, we collect polls from all the pollsters[16] as none performed frequent polling. We collect 65 polls for approximately 1 year of our data, which we expand to 98 polls by replicating polls to days when no polls were conducted in order to ensure a sufficient number of training points. The parties selected were: the Social Democratic Party (SPÖ), the People's Party (ÖVP), the Freedom Party (FPÖ) and the Green Alternative Party (GRÜ), which totalled over 90% of the voting intention at the beginning of our dataset. Thus we will have a prediction task for each party, three for the former setup and four for the later.

The voting intention time series are presented in Figures 4.1 and 4.2. We observe that the scores are quite stable across time, but a few change points exist. In the U.K. use case, LIB have an abrupt fall in popularity at the start of the time series and stabilise around of a score of 10%. CON and LAB interchange the first position in voting intention several times in the duration of the dataset. For Austria, the changes in voting intention are at the start of the time series between FPÖ and SPÖ and later between ÖVP and FPÖ. SPÖ and GRÜ have more stable scores.

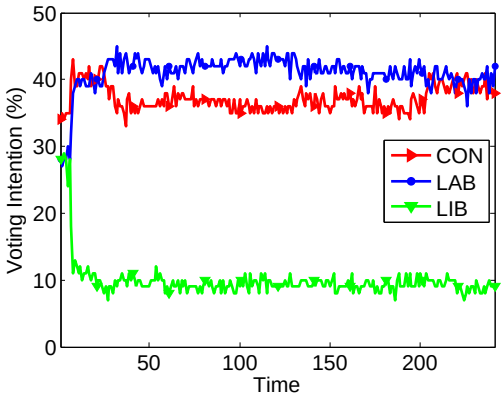


Figure 4.1: Voting intention for the U.K. experiments.

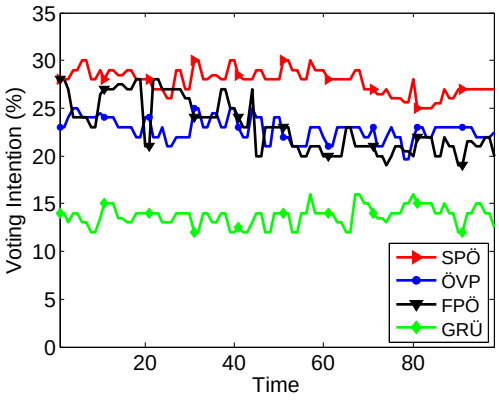


Figure 4.2: Voting intention for the Austria experiments.

In order to establish the usefulness of modeling voting intention with OSN data, we also compare to methods which use only historical voting intention scores. We define

[15] http://yougov.co.uk/

[16] http://de.wikipedia.org/wiki/Nationalratswahl_in_\%C3\%96sterreich_2013

Mean as the method that forecasts for a party its average voting intention the entire past data. **Last** is the method that forecasts the value of the most recent available poll. Due to the fact that polls are relatively stable in time, these method should achieve very good results. Any potential gain in accuracy beyond these methods would prove that OSN data is of significant usefulness for predicting voting intention.

4.3.2 Linear regression

As mentioned in Section 4.2.3, generalised linear models are a standard approach for regression tasks. In a regression task, a set of n input-response pairs $(\boldsymbol{x}_i, y_i)$ are given. The goal is to learn a function $f(x)$ that predicts y when only the value of $\boldsymbol{x}$ is given. Simple linear regression[17] learns a linear function of $\boldsymbol{x}$:

$$f(x) = \boldsymbol{x}^T\boldsymbol{w} + \beta \quad \text{where} \quad \boldsymbol{x}, \boldsymbol{w} \in \mathbb{R}^m \tag{4.1}$$

The objective is to learn the parameters of the model $\theta = (\boldsymbol{w}, \beta)$, where we refer to $\boldsymbol{w}$ as the feature weights and to β as the bias term. The goal is to minimise the loss function, which is the sum of a model-dependent loss function L and a regularisation penalty ψ. The loss function L is the sum of squared errors over the training set. Thus, the objective is:

$$\{\boldsymbol{w}, \beta\} = \underset{\boldsymbol{w},\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\boldsymbol{x}_i^T \cdot \boldsymbol{w} + \beta - y_i\right)^2 + \psi(\boldsymbol{w}, \rho) \tag{4.2}$$

Common choices for ψ are l_1 and l_2 regularisation (see Section 4.2.3). The l_1-norm regularisation has found many applications in several scientific fields as it encourages sparse solutions which reduce the possibility of overfitting and enhance the interpretability of the inferred model [Hastie et al., 2009]. This is especially the case when the number of training examples is less than the number of features, as usually with text features. We use the elastic net regulariser [Zou and Hastie, 2005], which applies an extra penalty on the l_2-norm of the weight vector. This can resolve instability issues of LASSO which arise when correlated predictors exist in the input data [Zhao and Yu, 2006]. Its regularisation function $\psi_{\mathbf{el}}$ is defined by:

$$\psi_{\mathbf{el}}(\boldsymbol{w}, \lambda, \alpha) = \lambda \left((1-\alpha)\frac{\|\boldsymbol{w}\|_2^2}{2} + \alpha\|\boldsymbol{w}\|_1 \right) \tag{4.3}$$

where $\lambda > 0$ and $\alpha \in [0, 1)$; setting parameter α to its extremes transforms elastic net to ridge regression ($\alpha = 0$) or LASSO ($\alpha = 1$).

[17] referred to as 'linear regression' further

In our setting, the number of samples n is represented by the number of polls used in training. We will consider word frequencies as features of the regression task. Corresponding to each poll, the frequencies are aggregated over the time interval between the previous poll and the current. We only use raw frequencies due to their simplicity in computation and interpretation. With a different use case, we have experimented with using deeper linguistic features, such as entities or part-of-speech tags [Preoţiuc-Pietro et al., 2014]. However, this only led to only minor improvements over unigram word frequencies.

4.3.3 Bilinear regression

As we have highlighted in Section 4.1.2, when using OSN data important issues arise from the vast volume of irrelevant messages to our prediction task. The presence of these can lead to arbitrary correlations in training and poor generalisation. In Section 2.1.2 we have seen that users of OSNs can be of different types, only a sub-category of which discuss current news and politics. This, together with demographic bias and opinion spam lead to the conclusion that not all users should be treated 'equal' in a predictive model. Rather than identifying and quantifying each of these sources of bias, a model should automatically account for them.

In this section we present a novel approach which performs high quality filtering automatically, through modelling not just words but also users. This is framed as a bilinear model with sparse regulariser [Lampos et al., 2013]. Regularised regression on both user and word spaces allows for an automatic selection of the most important words and users, performing at the same time an improved noise filtering.

In addition, we explore more advanced regularisation functions in multi-task learning schemes that can exploit shared structure in the feature space. The latter property becomes very useful in multi-output regression scenarios, where selected features are expected to have correlated or anti-correlated impact on each output (e.g. when inferring voting intentions for similar or competing political parties).

We highlight that our bilinear models can be used with any combination of two variables with which the data can be decomposed in the same manner. In the rest of this section we will refer to the two sets of variables as the 'words' and 'users'.

4.3.3.1 Bilinear framework

A number of different possibilities exist for incorporating user information into a regression model. A simple approach is to expand the feature set, such that each

user's effect on the response variable can be modelled separately. Although flexible, this approach would be doomed to failure due to the sheer size of the resulting feature set, and the propensity to overfit all but the largest of training sets. One solution is to group users into different types (e.g. journalist, politician, activist) but this presupposes a method for classification or clustering of users which is a non-trivial undertaking. Besides, these naïve approaches fail to account for the fact that most users use similar words to express their opinions, by separately parameterising the model for different users or user groups.

We propose to account for individual users while restricting all users to share the same vocabulary. This is formulated as a bilinear predictive model:

$$f(X) = \boldsymbol{u}^T X \boldsymbol{w} + \beta \tag{4.4}$$

where $\boldsymbol{u} \in \mathbb{R}^p$ are the user weights, $\boldsymbol{w} \in \mathbb{R}^m$ are the word weights, X is a $m \times p$ matrix of word-user frequencies and $\theta = (\boldsymbol{u}, \boldsymbol{w}, \beta)$ are the model parameters. Let $\mathcal{Q} \in \mathbb{R}^{n \times k \times p}$ be a tensor which captures our training inputs, where n is the number of samples, m is the number of words and p is the number of users; $\mathcal{Q}$ can simply be interpreted as n stacked versions of X (denoted by $\mathcal{Q}_i$), one for each training sample (i.e. time step in our setup). Each element $\mathcal{Q}_{ijk}$ holds the frequency of term j for user k during the time associated with sample i. If a user k has posted $c_{i \cdot k}$ messages during day i, and $c_{ijk} \leq c_{i \cdot k}$ of them contain a term j, then the frequency of j for this day and user is defined as $\mathcal{Q}_{ijk} = \frac{c_{ijk}}{c_{i \cdot k}}$.

Aiming to learn sparse sets of users and terms that are representative of the voting intention signal, we formulate our optimisation task as follows:

$$\{\boldsymbol{w}^*, \boldsymbol{u}^*, \beta^*\} = \underset{\boldsymbol{w}, \boldsymbol{u}, \beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\boldsymbol{u}^T \mathcal{Q}_i \boldsymbol{w} + \beta - y_i\right)^2 + \psi(\boldsymbol{w}, \rho_1) + \psi(\boldsymbol{u}, \rho_2) \tag{4.5}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the response variable, $\boldsymbol{u}^{\mathrm{T}} \mathcal{Q}_i \boldsymbol{w}$ expresses the bilinear term, $\beta \in \mathbb{R}$ is a bias term and $\psi(\cdot)$ is a regularisation function over both the user and word weights with different parameters ρ_1 and ρ_2. The first term in Equation 4.5 is the standard sum squared error. In the main formulation of our bilinear model, as the regularisation function ψ we again use the elastic net [Zou and Hastie, 2005].

The optimisation of the objective in Equation 4.5 can be treated as a biconvex learning task [Al-Khayyal and Falk, 1983], by observing that for a fixed $\boldsymbol{w}$, learning $\boldsymbol{u}$ is a convex problem and vice versa. Biconvex functions and possible applications have been well studied in the optimisation literature [Quesada and Grossmann, 1995,

Pirsiavash et al., 2009]. Their main advantage is the ability to solve efficiently non-convex problems by a repeated application of two convex processes i.e. a form of coordinate ascent. In our case, the bilinear technique makes it possible to explore both word and user spaces, while maintaining a reduced complexity: twice the number of iterations times that of the linear learner. The method for optimisation we use is FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) [Beck and Teboulle, 2009].

Therefore, in our bilinear approach we divide learning in two phases, where we learn word and user weights respectively. For the first phase we produce the term-scores matrix $\mathcal{V} \in \mathbb{R}^{n \times m}$ with elements given by:

$$\mathcal{V}_{ij} = \sum_{z=1}^{p} u_z \mathcal{Q}_{ijz}. \tag{4.6}$$

$\mathcal{V}$ contains weighted sums of term frequencies over all users for the considered set of days. The weights are held in $\boldsymbol{u}$ and are representative of each user. The initial optimisation task is formulated as:

$$\{\boldsymbol{w}^*, \beta^*\} = \underset{\boldsymbol{w},\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathcal{V}\boldsymbol{w}_i + \beta - \boldsymbol{y}_i\right)^2 + \psi\left(\boldsymbol{w}, \rho_1\right) \tag{4.7}$$

where we aim to learn a sparse but consistent vector of weights $\boldsymbol{w}^*$ for the words of our vocabulary.

In the second phase, we are using $\boldsymbol{w}^*$ to form the user-scores matrix $\mathcal{D} \in \mathbb{R}^{n \times p}$:

$$\mathcal{D}_{ik} = \sum_{z=1}^{m} w_z^* \mathcal{Q}_{izk} \tag{4.8}$$

which now contains weighted sums over all terms for the same set of days. The optimisation task becomes:

$$\{\boldsymbol{u}^*, \beta^*\} = \underset{\boldsymbol{u},\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathcal{D}\boldsymbol{u}_i + \beta - \boldsymbol{y}_i\right)^2 + \psi\left(\boldsymbol{u}, \rho_2\right) \tag{4.9}$$

By inserting the weights of the second phase back to phase one, we can iterate the process as in each step we are dealing with a convex problem. We cannot claim that a global optimum will be reached, but biconvexity guarantees that our global objective (Equation 4.5) will decrease in each step of this iterative process (i.e. local optimality). We will refer to this method as Bilinear Elastic Net (**BEN**).

4.3.3.2 Exploiting word-target or user-target relationships

The previous model assumes that the response variable $\boldsymbol{y}$ holds information about a single inference target. However, the task that we are addressing in this paper usually implies the existence of several targets, i.e. different political parties. An important property, therefore, is the ability to perform multiple output regression. A simple way of adapting the model to the multiple output scenario is by framing a separate learning problem for each output, but tying together some of the parameters. Here we consider tying together the user weights $\boldsymbol{u}$, to enforce that the same set of users are relevant to all tasks, while learning different word weights. Note that the converse situation, where $\boldsymbol{w}$'s are tied and $\boldsymbol{u}$'s are independent, can be formulated in an equivalent manner.

Suppose that our target variable $\boldsymbol{Y} \in \mathbb{R}^{\tau \cdot n}$ refers now to τ different response variables (e.g. political parties), $\boldsymbol{Y} = \left[\boldsymbol{y}_1^{\mathrm{T}} \boldsymbol{y}_2^{\mathrm{T}} ... \boldsymbol{y}_\tau^{\mathrm{T}}\right]^{\mathrm{T}}$; in this formation the top n elements of $\boldsymbol{Y}$ match to the first response variable, the next n elements to the second and so on. In the first phase of the bilinear model, we would have to solve the following optimisation task:

$$\{\boldsymbol{w}^*, \beta^*\} = \underset{w,\beta}{\operatorname{argmin}} \sum_{i=1}^{\tau} \|\mathcal{V}\boldsymbol{w_i} + \beta_i - \boldsymbol{y}_i\|_2^2 + \sum_{i=1}^{\tau} \psi\left(\boldsymbol{w}_i, \rho_1\right) \tag{4.10}$$

where $\mathcal{V}$ is given by Equation 4.6 and $\boldsymbol{w}^* \in \mathbb{R}^{\tau m}$ denotes the vector of word weights which can be sliced into τ sub-vectors $\{\boldsymbol{w}_1^*, ..., \boldsymbol{w}_\tau^*\}$ each one representing a response variable. In the second phase, sub-vectors $\boldsymbol{w}_i^*$ are used to form the input matrices $\mathcal{D}_i$, $i \in \{1, ..., \tau\}$ with elements given by Equation 4.8. The input matrix $\mathcal{D}'$ is formed by the vertical concatenation of all $\mathcal{D}_i$ user score matrices i.e. $\mathcal{D}' = \left[\mathcal{D}_1^{\mathrm{T}} \ ... \ \mathcal{D}_\tau^{\mathrm{T}}\right]^{\mathrm{T}}$, and the optimisation target is equivalent to the one expressed in Equation 4.9. Since $\mathcal{D}' \in \mathbb{R}^{\tau \cdot n \times p}$, the user weight vector $\boldsymbol{u}^* \in \mathbb{R}^p$ and thus, we are learning a single weight per user and not one per response variable as in the previous step.

The method described above allows learning different word weights per response variable and then binds them under a shared set of user weights. As mentioned before, one could also try the opposite and learn a single weight per word and different user weights for each response variable. In this case, the first phase of the model would have to solve:

$$\{\boldsymbol{u}^*, \beta^*\} = \underset{u,\beta}{\operatorname{argmin}} \sum_{i=1}^{\tau} \|\mathcal{D}\boldsymbol{u_i} + \beta_i - \boldsymbol{y}_i\|_2^2 + \sum_{i=1}^{\tau} \psi\left(\boldsymbol{u}_i, \rho_2\right) \tag{4.11}$$

where $\mathcal{D}$ is given by Equation 4.8 and $\boldsymbol{u}^* \in \mathbb{R}^{\tau p}$ denotes the vector of user weights which can be sliced into τ sub-vectors $\{\boldsymbol{u}_1^*, ..., \boldsymbol{u}_\tau^*\}$ each one representing a response

variable. In the second phase, sub-vectors $\boldsymbol{u}_i^*$ are used to form the input matrices $\mathcal{V}_i$, $i \in \{1, ..., \tau\}$ with elements given by Equation 4.6. The input matrix $\mathcal{V}'$ is formed by the vertical concatenation of all $\mathcal{V}_i$ word score matrices i.e. $\mathcal{V}' = \left[\mathcal{V}_1^{\mathrm{T}} \ ... \ \mathcal{V}_\tau^{\mathrm{T}}\right]^{\mathrm{T}}$, and the optimisation target is equivalent to the one expressed in Equation 4.9.

Both those models can also be optimised in an iterative process. However, experiments revealed that those approaches did not improve on the performance of the BEN model. Still, this behaviour could be problem-specific, i.e. learning different words from a shared set of users (and the opposite) may not be a good modelling practice for this task. Nevertheless, this observation served as a motivation for the method described in the next section, where we extract a consistent set of words and users that are weighted differently among the considered political entities.

4.3.3.3 Multi-task learning

All previous models – even when combining all inference targets – were not able to explore relationships across the different task domains; in our case, a task domain is defined by a specific political party. Ideally, we would like to make a sparse selection of words and users but with a regulariser that promotes inter-task sharing of structure, so that many features may have a positive influence towards one or more targets, but negative towards the remaining ones. It is possible to achieve this multi-task learning property by introducing a different set of regularisation constraints in the optimisation function.

We perform multi-task learning using an extension of group LASSO [Yuan and Lin, 2006], a method known as l_1/l_2 regularisation [Argyriou et al., 2007]. Group LASSO exploits a predefined group structure on the feature space and tries to achieve sparsity in the group-level i.e. it does not perform feature selection (unlike the elastic net), but group selection. The l_1/l_2 regulariser extends this notion for a τ-dimensional response variable. The global optimisation target is now formulated as:

$$\{W^*, U^*, \boldsymbol{\beta}^*\} = \underset{W,U,\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{t=1}^{\tau} \sum_{i=1}^{n} \left(\boldsymbol{u}_t^{\mathrm{T}} \mathcal{Q}_i \boldsymbol{w}_t + \beta_t - y_{ti}\right)^2 + \lambda_1 \sum_{j=1}^{m} \|W_j\|_2 + \lambda_2 \sum_{k=1}^{p} \|U_k\|_2 \tag{4.12}$$

where the input matrix $\mathcal{Q}_i$ is defined in the same way as earlier, $W = [\boldsymbol{w}_1 \ ... \ \boldsymbol{w}_\tau]$ is the word weight matrix (each $\boldsymbol{w}_t$ refers to the t-th task i.e. political entity), equivalently $U = [\boldsymbol{u}_1 \ ... \ \boldsymbol{u}_\tau]$, W_j and U_j denote the j-th rows of weight matrices W and U respectively, and vector $\boldsymbol{\beta} \in \mathbb{R}^\tau$ holds the bias terms per task. We are now regularising the $\ell_{2,1}$ mixed norm of W and U, which is defined as the sum of the row ℓ_2-norms for

those matrices. In this optimisation process, we aim to enforce structured sparsity in the feature space, biasing user and word weights accross all tasks to a value of zero [Argyriou et al., 2007, Eisenstein et al., 2011]. Only a subset of both users and words will have non-zero weights, the same across all tasks (corresponding to the rows of W and U). Consequently, we are essentially performing filtering while assigning weights to the selected users or words. This is suitable for many use cases, such as our political application, where pro-A often means anti-B.

Equation 4.12 can be split into two convex tasks (following the same notions as in Equations. 4.7 and 4.9), where we individually learn $\{W, \boldsymbol{\beta}\}$ and then $\{U, \boldsymbol{\beta}\}$; each step of the process is a standard linear regression problem with an ℓ_1/ℓ_2 regulariser. Again, we are able iterate this bilinear process and in each step convexity is guaranteed. We refer to this method as Bilinear Group ℓ_1/ℓ_2 (**BGL**).

4.3.4 Experimental setup

Basic pre-processing has been applied on the vocabularies of both the U.K. Users and Austrian Politics datasets aiming to filter out some of the word features and partially reduce the dimensionality of the problem. Stop words and URLs were removed from both dictionaries, together with character sequences of length <4 and <3 respectively.[18] As the vocabulary size of the former dataset was significantly larger, for this data set we have additionally merged Twitter hashtags with their exact non topic word match, where possible (by dropping the '#' when the word existed in the index). After performing the pre-processing routines described above, the vocabulary sizes for were 80,976 and 22,917 respectively. The number of users is 42,484 in the U.K. scenario and 1,118 in the Austrian one. In building the feature matrices X_i we use the messages since the day of the previous poll up to the day of the current poll. We could have considered a lag in response to a poll, as in [O'Connor et al., 2010], but decided not to because the polls have good frequency and their fieldwork dates are specified.

To evaluate the predictive accuracy of our methods, we emulate a real-life scenario of voting intention prediction. The evaluation process starts by using a fixed set of polls matching to consecutive time points in the past for training and validating the parameters of each model. Testing is performed on the following δ (unseen) polls of the data set. In the next step of the evaluation process, the training/validation set is increased by merging it with the previously used test set (δ polls), and testing is now

[18] Most of the times those character sequences were not valid words. This pattern was different in each language and thus, a different filtering threshold was applied in each data set.

performed on the next δ unseen polls. In our experiments, the number of steps in this evaluation process is set to 10 and in each step the size of the test set is set to $\delta = 5$ polls. Hence, each model is tested on 50 unseen and consecutive in time samples. The loss function in our evaluation is the standard Mean Square Error (**MSE**), but to allow a better interpretation of the results, we display its root (**RMSE**) in tables and figures.[19] RMSE is computed per party in each fold and the reported result is the average across folds.

The hyperparameters of each model (α_i for BEN and λ_i for BEN and BGL, $i \in \{1, 2\}$) are optimised using a held-out validation set by performing grid search. Note that it may be tempting to adapt the regularisation parameters in each phase of the iterative training loop, however this would change the global objective (see Equations 4.5 and 4.12) and thus convergence will not be guaranteed. A key question is how many iterations of training are required to reach convergence. Figure 4.3 illustrates how the BEN global objective function (Equation 4.5) converges during this iterative process and the model's performance on an unseen test set. Notice that there is a large performance improvement after the first step (which alone is a linear solver), but overfitting occurs after step 11. Based on this result, for subsequent experiments we run the training process for two iterations (4 steps), and take the best performing model on the held-out validation set.

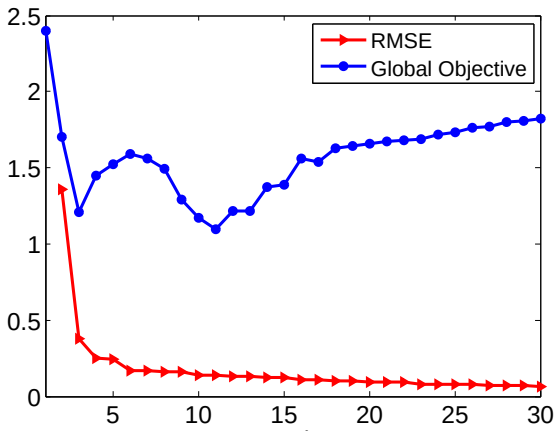


Figure 4.3: Global objective function and RMSE on a validation set for BEN in 15 iterations (30 steps) of the model.

[19]RMSE has the same metric units as the response variable.

4.3.5 Quantitative results

Forecasting results comparing inferred voting intention percentages and polls are presented for both datasets in Tables 4.1 and 4.2 respectively. For the U.K. study, both BEN and BGL perform best in average performance across all parties. However in the Austrian case study, LEN performs better that BEN, something that could be justified by the fact that the number of users in this dataset was small enough and they were chosen to be relevant for current affairs. Imposing a filtering on them is thus not needed. Nevertheless, the difference in performance was rather small (≈ 0.26 error) and the inferences of LEN and BEN followed a very similar pattern ($\bar{\rho} = .94$ with $p < 10^{-10}$).[20] Multi-task learning (BGL) delivered the best inference performance in both case studies, which was on average smaller than 1.48 (RMSE).

Inferences for both BEN and BGL are been plotted in Figures 4.6, 4.7, 4.8 and 4.9. They are presented as continuous lines of 50 inferred points (per party) which are created by concatenating the inferences on all test sets. For the U.K. case study, one may observe that BEN (Figure 4.6) cannot register any change in leading voting intention supremacy (CON versus LAB) with the exception of one test point; BGL (Figure 4.8) performs much better in that aspect. In the Austrian case study this characteristic becomes more obvious. BEN (Figure 4.7) consistently predicts the wrong ranking of ÖVP and FPÖ, whereas BGL (Figure 4.9) does much better. Most importantly, a general observation is that BEN's predictions are very smooth. This might be a result of overfitting the model to a single response variable which usually has a smooth behaviour. On the contrary, the multi-task learning property of BGL reduces this type of overfitting providing more statistical evidence for the terms and users and thus, yielding not only a better inference performance, but also a more accurate model. We note that further investigation on the validity of the results is necessary in order to make this tool useful in practice and to replace, at least partially, the need of traditional voting intention polling.

4.3.6 Qualitative analysis

In this section, we refer to features that have been selected and weighted as significant by our bilinear learning models. The top positive weighted words for each party are presented in word clouds in Figures 4.10, 4.11 and 4.12. We notice first that key political terms are selected with high weights (e.g. tory, labour). Some words are related to events which impact political outcomes (e.g. egypt, election), while others

[20] Pearson's linear correlation averaged across the four Austrian parties.

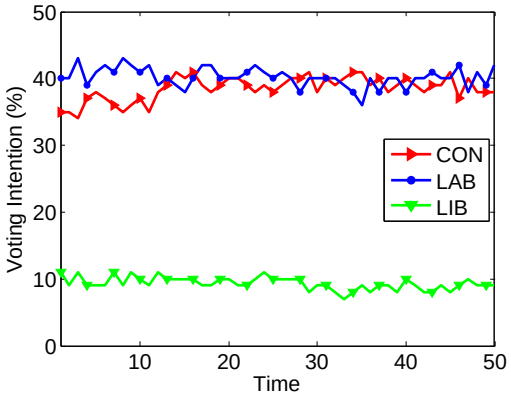


Figure 4.4: Voting intention for the U.K. tests.

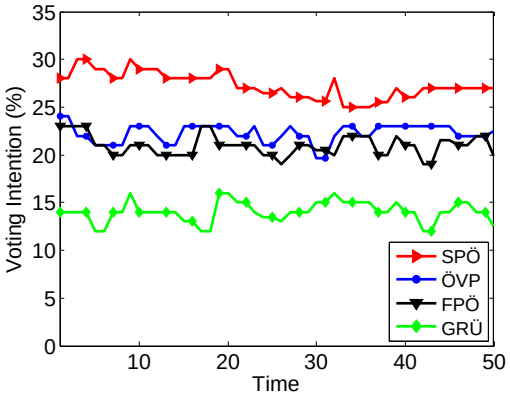


Figure 4.5: Voting intention for the Austria tests.

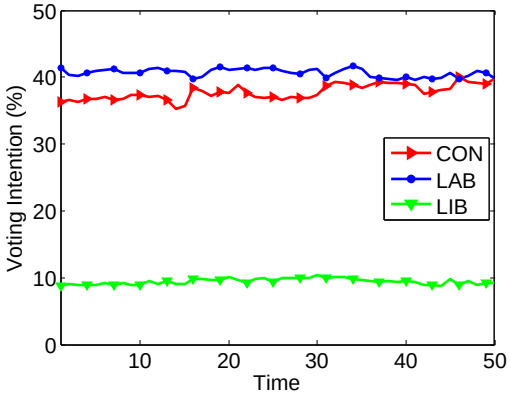


Figure 4.6: Results of the BEN model for the U.K.

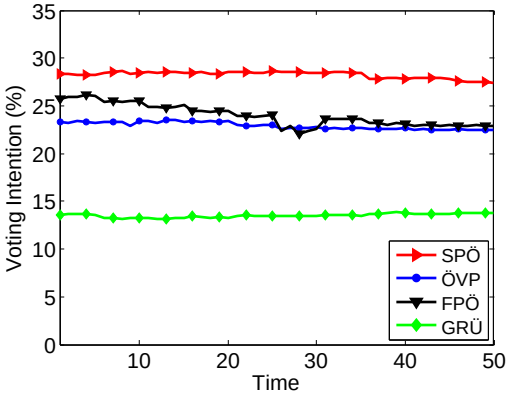


Figure 4.7: Results of the BEN model for Austria.

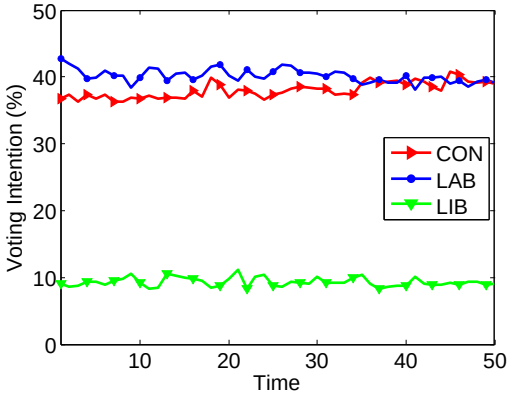


Figure 4.8: Results of the BGL model for the U.K.

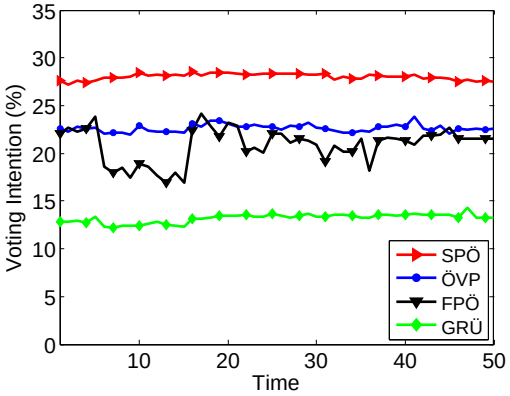


Figure 4.9: Results of the BGL model for Austria.

	CON	**LAB**	**LIB**	$\boldsymbol{\mu}$
Mean	2.272	1.663	1.136	1.69
Last	2	2.074	1.095	1.723
LEN	3.845	2.912	2.445	3.067
BEN	1.939 ± 1.92	1.644 ± 1.59	1.136 ± 1.03	1.573
BGL	$\mathbf{1.785} \pm 1.53$	$\mathbf{1.595} \pm 1.09$	$\mathbf{1.054} \pm 0.81$	**1.478**

Table 4.1: U.K. case study – Average RMSEs representing the error of the inferred voting intention percentage for the 10-step validation process; $\boldsymbol{\mu}$ denotes the mean RMSE across the three political parties for each baseline or inference method. Standard deviation across folds shown for the bilinear models.

	SPÖ	**ÖVP**	**FPÖ**	**GRÜ**	$\boldsymbol{\mu}$
Mean	1.535	1.373	3.3	1.197	1.851
Last	**1.148**	1.556	**1.639**	1.536	1.47
LEN	1.291	1.286	2.039	**1.152**	1.442
BEN	1.392 ± 1.39	1.31 ± 1.00	2.89 ± 2.20	1.205 ± 1.05	1.699
BGL	1.619 ± 1.54	$\mathbf{1.005} \pm 0.90$	1.757 ± 1.39	1.374 ± 1.10	**1.439**

Table 4.2: Austrian case study – Average RMSEs for the 10-step validation process. Standard deviation across folds shown for the bilinear models.

(e.g. video, blog, live) are potentially indicative of the type of tweets which contain news or opinions which may impact political voting intention. While the word clouds are representative, they are not exhaustive, as the sparse regulariser aims to remove words which have a correlated behaviour. The selected users can not be displayed for privacy reasons, but these mostly report news or comment on politics and current affairs, which is consistent to our goal of filtering out users which are uninformative to our regression goal.

Based on the weights for the word and the user spaces that we retrieve after the application of BGL in the last step of the evaluation process (see the previous section), we compute a score (weighted sum) for each tweet in our training datasets. Table 4.3 shows examples of interesting tweets amongst the top weighted ones (positively as well as negatively) per party. Together with their text and scores, we also provide an attribute for the author (anonymised for privacy reasons). In the displayed tweets for the U.K. study, the only possible outlier is the 'Art Fanzine'; still, it seems to register a consistent behaviour (positive towards LAB, negative towards LIB) and, of course, hidden, indirect relationships may exist between political opinion and art. The Austrian case study revealed even more interesting tweets since training was

conducted on data from a selected set of users. For a better interpretation of the presented tweets, it may be useful to know that 'Johannes Voggenhuber' (who receives a positive comment for his book) and 'Peter Pilz' (whose comment is questioned) are members of GRÜ, 'Krone' (or Kronen Zeitung) is the major newspaper in Austria[21] and that FPÖ is labelled as a far right party, something that may cause various reactions from 'Human Rights' organisations.

Figure 4.10: Highest positive weighted words for the Conservative Party (CON) (final fold). Font size is proportional to the word weight.

Figure 4.11: Highest positive weighted words for the Liberal-Democrat (LIB) Party (final fold). Font size is proportional to the word weight.

Figure 4.12: Highest positive weighted words for the Labour Party (LAB) (final fold). Font size is proportional to the word weight.

[21] 'Accused of abusing its near monopoly to manipulate public opinion in Austria', Wikipedia, 19/02/2013, `http://en.wikipedia.org/wiki/Kronen_Zeitung`

Party	Tweet	Score	Author
CON	PM in friendly chat with top EU mate, Sweden's Fredrik Reinfeldt, before family photo	1.334	Journalist
	Have Liberal Democrats broken electoral rules? Blog on Labour complaint to cabinet secretary	−0.991	Journalist
LAB	Blog Post Liverpool: City of Radicals Website now Live <*link*> #liverpool #art	1.954	Art Fanzine
	I am so pleased to hear Paul Savage who worked for the Labour group has been Appointed the Marketing manager for the baths hall GREAT NEWS	−0.552	Politician (Labour)
LIB	RT *@user*: Must be awful for TV bosses to keep getting knocked back by all the women they ask to host election night (via *@user*)	0.874	LibDem MP
	Blog Post Liverpool: City of Radicals 2011 – More Details Announced #liverpool #art	−0.521	Art Fanzine
SPÖ	Inflationsrate in Ö. im Juli leicht gesunken: von 2,2 auf 2,1%. Teurer wurde Wohnen, Wasser, Energie.	0.745	Journalist
	Hans Rauscher zu Felix #Baumgartner "A klaner Hitler" <*link*>	−1.711	Journalist
ÖVP	#IchPirat setze mich dafür ein, dass eine große Koalition _mathematisch_ verhindert wird! 1.Geige: #Gruene + #FPOe + #OeVP	4.953	User
	kann das buch "res publica" von johannes #voggenhuber wirklich empfehlen! so zum nachdenken und so... #europa #demokratie	−2.323	User
FPÖ	Neue Kampagne der #Krone zur #Wehrpflicht: "GIB BELLO EINE STIMME!"	7.44	Political satire
	Kampagne der Wiener SPÖ "zum Zusammenleben" spielt Rechtspopulisten in die Hände <*link*>	−3.44	Human Rights
GRÜ	Protestsong gegen die Abschaffung des Bachelor-Studiums Internationale Entwicklung: <*link*> #IEbleibt #unibrennt #uniwut	1.45	Student Union
	Pilz "ich will in dieser Republik weder kriminelle Asylwerber, noch kriminelle orange Politiker" - BZÖ-Abschiebung ok, aber wohin? #amPunkt	−2.172	User

Table 4.3: Examples of tweets with top positive and negative scores per party for both datasets.

4.4 Non-stationary models of forecasting

Experiments and analysis of results from the previous Section have shown that text in social media correlates to real-world indicators. However, the forecasting task is of a streaming nature, something which was not exploited by the batch algorithms used to train the models we presented. The prediction model needs to be updated as new data arrives one by one in a (possibly) unbounded stream containing the word × user frequency matrices $\mathcal{Q}_i$ and the voting intentions for the parties y_i. Online learning, presented in Section 4.2.1, is usually used in this setting in order to avoid retraining the full model to incorporate each input, which is usually very costly. This class of methods are particularly useful when the distribution of the data is non-stationary i.e. its properties change with time. In our case, we study the impact of considering newer data to play a more important role in predictions in comparison to older data.

4.4.1 Online bilinear learning

We start by introducing an online biconvex learning scheme for the bilinear models in Section 4.3.3. This allows the bilinear models to work in a streaming, one pass setting.

This allows the model to update the weights for words, users as well as the bias terms on the addition of new data. This scheme uses Stochastic Gradient Descent (SGD) with proximal updates for regularisation. Our experiments (Section 4.4.1.2) use the online version of the BGL model, as this obtained the best results in the batch setting. We will analyse *regret* by comparing the results with those of the batch model.

4.4.1.1 Algorithm

Our technique extends Stochastic Gradient Descent methods with proximal updates for regularisation to work in a biconvex context. The objective is to optimise the loss function e.g. Equation 4.12 for BGL. Optimisation is usually performed via gradient descent.

Gradient descent methods consist on iteratively following the gradient of the differentiable objective function F. If this is also convex then reaching a global optimum is guaranteed, otherwise the optimum might be only local. Assuming differentiable and decomposable F, the gradient of F at w is given by $\nabla F(w) = \frac{1}{N}\sum_{n=1}^{N}\nabla F_n(w)$. An algorithm for optimising $F(w)$ iteratively updates the weight vector w following the negative of the gradient by the learning rate η: $w^* = w - \eta\nabla F(w)$. The downside is that the gradient needs to be recomputed using the entire dataset. When the number or test points or the dimensionality of the problem is large, the computational cost is high, making learning new models from scratch costly.

An alternative to gradient descent is represented by Stochastic Gradient Descent [Bottou, 2004]. This replaces the full gradient computation $\nabla F(w)$ with an estimate of the gradient built from a small subset of the training instances, T.For online learning, T is chosen to contain only the most recent timestamp before each prediction. Stochastic Gradient Descent converges with probability one or almost surely [Rogers and Williams, 2000] to a minimum (global in the case of convex objectives) with the learning rate $\eta_t = \mathcal{O}(1/\sqrt{t})$.

However, Stochastic Gradient Descent has issues when seeking sparse solutions, as those promoted by the ℓ_1 and ℓ_1/ℓ_2 regularisers used in our bilinear models. Since the regularisers are non-differentiable at the origin, the solutions are rarely sparse, having components close, but not equal to zero [Martins et al., 2011b]. These solutions are slightly inferior and are a consequence of noisy SGD. In order to address this problem we follow the method from [Martins et al., 2011a]. The (sub)gradient is computed of F only with respect to the loss function, followed by proximal steps for each component of the regulariser. The proximal operator $\text{prox}_\phi(w)$ [Moreau, 1962] gives

Algorithm 1 Online Biconvex Proximal Gradient Algorithm.

$\mathcal{Q} \leftarrow \{\mathcal{Q}_1, ..., \mathcal{Q}_n\}$ ▷ A streaming sequence of word × user frequency matrices
$Y \leftarrow \{y_1, ..., y_n\}$ ▷ A streaming sequence of response variables
$\Theta \leftarrow \{\eta, d_w, d_u\}$ ▷ Learning rates, proximal parameters
procedure OnlineBilinearLearner($\mathcal{Q}$, Y, Θ)
  initialise $\boldsymbol{w}$ ▷ Initialise word, user weights and bias
  for $\langle \mathcal{Q}_i, y_i \rangle \in \langle \mathcal{Q}, Y \rangle$ **do**
    while !StoppingCondition(Θ; $\boldsymbol{w}, \boldsymbol{u}, \beta$) **do**
      $\mathcal{V}^i \leftarrow \boldsymbol{u}^T \mathcal{Q}_i + \beta$ ▷ The word-scores matrix
      $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta_t \nabla L(\boldsymbol{w}; \mathcal{V}_i, y_i, \{\boldsymbol{w}, \beta\})$ ▷ Word Gradient Step
      $\boldsymbol{w} = prox(\boldsymbol{w})$ ▷ Word Proximal Step
      $\mathcal{D}_i \leftarrow \mathcal{Q}_i \boldsymbol{w} + \beta$ ▷ The user-scores matrix
      $\boldsymbol{u} \leftarrow \boldsymbol{u} - \eta_t \nabla L(\boldsymbol{u}; \mathcal{D}_i, y_i, \{\boldsymbol{u}, \beta\})$ ▷ User Gradient Step
      $\boldsymbol{u} = prox(\boldsymbol{u})$ ▷ User Proximal Step
      $\mathcal{B}_i \leftarrow \boldsymbol{u}^{\mathrm{T}} \mathcal{Q}_i \boldsymbol{w} + \beta$ ▷ Current estimate with bias
      $\beta \leftarrow \beta - \eta_t \nabla L(\beta; \mathcal{B}_i, y_i, \{\beta\})$ ▷ Bias Gradient Update
    end while
  end for
end procedure

the best trade-off between a good approximation of w and the cost associated with the function ϕ:

$$\text{prox}_\phi(w) = \underset{w'}{\text{argmin}} \frac{1}{2}\|w' - w\|^2 + \phi(w') \tag{4.13}$$

In our algorithm, ϕ is the regulariser and w are the current weights as obtained by the gradient descent step. The weights thus are again subject to sparsity constraints.

Algorithm 1 outlines the generic algorithm for online learning using Stochastic Gradient Descent with proximal updates for bilinear models. This works as follows. For each training pair $\langle \mathcal{Q}_i, y_i \rangle$, the algorithm alternatively updates each of the two sets of weights ($\boldsymbol{u}$ or $\boldsymbol{w}$) whilst keeping the other fixed. Similarly to the batch setting, this is performed for a number of iterations. The update step consists, for any of the two sets of weights, in alternating between (sub)gradient steps with respect to the loss function and proximal steps with respect to the regulariser. At the end of each iteration, a gradient update on the bias is performed by keeping both sets of weights fixed. In our generic algorithm, the loss function and the regulariser are not explicitly specified. Any loss function with a calculable (sub)gradient can be used in this setting. Analogously, the regulariser needs to have a calculable proximal operator.

The best experimental results in Section 4.3.4 were obtained using the BGL model. This uses as loss function the sum squared loss. The loss for a single training pair

$\langle \mathcal{Q}_i, y_i \rangle$ is:

$$L(\mathcal{Q}_i, y_i; \{\boldsymbol{w}, \boldsymbol{u}, \boldsymbol{\beta}\}) = \sum_{t=1}^{\tau} \left(\boldsymbol{u}_t^{\mathrm{T}} \mathcal{Q}_i \boldsymbol{w}_t + \beta_t - y_{ti}\right)^2 \tag{4.14}$$

When updating $\boldsymbol{w}$, we first calculate $\mathcal{V}_i$, thus keeping $\boldsymbol{u}$ fixed. The gradient of this loss function is used to update $\boldsymbol{w}$:

$$\begin{aligned} \mathcal{V}_i =& \boldsymbol{u}^T \mathcal{Q}_i \\ L(\mathcal{V}_i, y_i, \{\boldsymbol{w}, \beta\}) =& \sum_{t=1}^{\tau} \left(\mathcal{V}_i \boldsymbol{w}_t + \beta - y_{ti}\right)^2 \\ \nabla L(\boldsymbol{w}; \mathcal{V}_i, y_i, \{\boldsymbol{w}, \beta\}) =& \sum_{t=1}^{\tau} 2\mathcal{V}_i \left(\mathcal{V}_i \boldsymbol{w}_t + \beta - y_{ti}\right) \end{aligned} \tag{4.15}$$

This is a convex function [22] and thus updating following the gradient is guaranteed to lead to a global optima. The quantity by which the gradient is followed is a parameter of the online algorithm η_t. The gradient update for $\boldsymbol{w}$ is computed in a similar way:

$$\begin{aligned} \mathcal{D}_i =& \mathcal{Q}_i \boldsymbol{w} \\ L(\mathcal{D}_i, y_i, \{\boldsymbol{u}, \beta\}) =& \sum_{t=1}^{\tau} \left(\boldsymbol{u}^T \mathcal{D}_i + \beta - y_{ti}\right)^2 \\ \nabla L(\boldsymbol{u}; \mathcal{D}_i, y_i, \{\boldsymbol{u}, \beta\}) =& \sum_{t=1}^{\tau} 2\mathcal{D}_i \left(\boldsymbol{u}^T \mathcal{D}_i + \beta - y_{ti}\right) \end{aligned} \tag{4.16}$$

Finally, we also update the bias term β. Its gradient is:

$$\begin{aligned} \mathcal{B}_i =& \boldsymbol{u}^{\mathrm{T}} \mathcal{Q}_i \boldsymbol{w} \\ L(\mathcal{B}_i, y_i, \{\beta\}) =& \sum_{t=1}^{\tau} \left(\mathcal{B}_i + \beta - y_{ti}\right)^2 \\ \nabla L(\beta; \mathcal{B}_i, y_i, \{\beta\}) =& \sum_{t=1}^{\tau} 2 \left(\mathcal{B}_i + \beta - y_{ti}\right) \end{aligned} \tag{4.17}$$

For regularisation, BGL uses the ℓ_1/ℓ_2 regulariser. The proximal update step for the words which form a group m is:

$$\mathrm{prox}_{GL}(\boldsymbol{w})_m = \begin{cases} 0 & \text{if } \|\boldsymbol{w}_m\|_2 \leq d_w \\ \frac{\|\boldsymbol{w}_m\|_2 - d_w}{\|\boldsymbol{w}_m\|_2} \boldsymbol{w}_m & \text{otherwise} \end{cases} \tag{4.18}$$

Practically, for the ℓ_1/ℓ_2 (and also ℓ_1) regulariser, this operator shrinks all values and pushes to zero those under a threshold, thus finding sparse solutions.

[22] The initial bilinear loss function in Equation 4.14 is non-convex

Algorithm 1 receives an input stream of training pairs $\{\langle Q_1, y_1\rangle, ..., \langle Q_n, y_n\rangle\}$. The algorithm has various parameters including initialisation, learning rates and regularisation parameters. First, before data is seen, the $\boldsymbol{w}$ and $\boldsymbol{u}$ vectors need to be initialised (Line 5). We found the convergence of the algorithm is very dependent on a good initialisation. Zero initialisation leads to zero gradient update, while a random one slows the rate of learning. In our experiments we initialised $\boldsymbol{u}$ uniformly with a low value (i.e. 0.1) and initialised $\boldsymbol{w}$ and the bias terms β with zero, making the first update of $\boldsymbol{w}$ to be data driven. We set the number of iterations to a minimum of 3 and maximum of 10, further stopping in case of few changes in the weights (i.e. $(\Delta\boldsymbol{w} + \Delta\boldsymbol{u} + \Delta\beta)/(\boldsymbol{w} + \boldsymbol{u} + \beta) < 0.01$) (Line 7).

4.4.1.2 Experiments

In this section we experiment with the online biconvex learner used to train a bilinear model. We aim to demonstrate the capabilities of the online learner by comparing it to the batch learner using the same experimental setup. Our experiments focus on the Austrian political use case as described in Section 4.3.4.

Regret analysis [Shalev-Shwartz, 2007] compares the cumulative loss of the sequentially trained online learning algorithm to the results of a batch model trained from scratch from all the data available before each prediction. Regret thus measures the penalty paid by the online model compared to the batch model. Our batch experiments simulated a real world scenario where predictions were made up to five time steps into the future (given the observations for these days). After the predictions were made, a new model was trained on the entire previous data and the new data points. We simulate the same setting for the online model. The model is trained online, makes the next five predictions and only after, the new data is incorporated into the model by online updates.

	SPÖ	**ÖVP**	**FPÖ**	**GRÜ**	$\boldsymbol{\mu}$
BGL Batch	1.619 ± 1.54	1.005 ± 0.90	1.757 ± 1.39	1.374 ± 1.10	1.439
BGL Online	2.028 ± 1.75	1.602 ± 1.01	1.722 ± 1.75	1.2465 ± 1.03	1.648

Table 4.4: Austrian online learning comparison – Average RMSEs for the 10-step validation process.

The results of the online model are presented in Table 4.4 and in Figure 4.13. We notice that the predictive performance of the online model is, as expected, worst than the batch model but only by around 0.2 RMSE.

Regret is plotted in Figure 4.14. We also give a measure of average regret as the mean of the differences between the batch and online model up to a given point in Figure 4.15. We measured the regret between the 10 testing steps. We observe that regret quickly increases in the first testing steps, with an decrease in the later steps. As more data is seen, the online model performs better predictions, reflected by a decrease in average regret. In steps 7 and 8 the online learner performs even slightly better than the batch algorithm, highlighted by a decrease in regret. This indicates possible non-stationarity of our data.

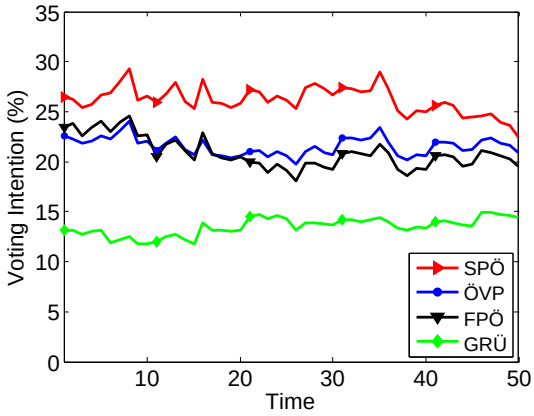


Figure 4.13: Results of the Online BGL model for Austria.

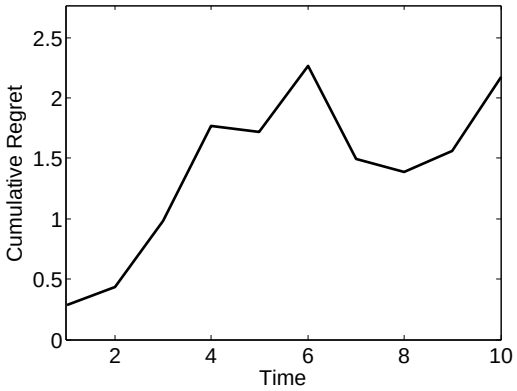


Figure 4.14: Regret (cumulative loss) over the 10 testing steps.

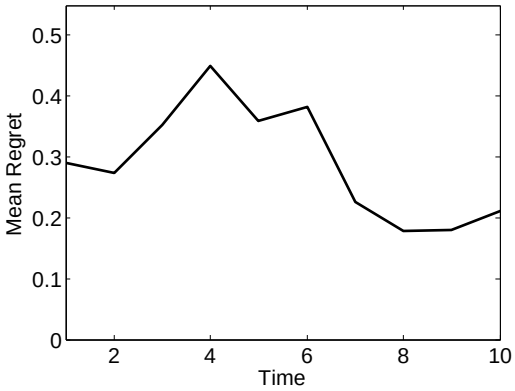


Figure 4.15: Average regret over the 10 testing steps.

4.4.2 Non-stationarity

In this chapter we examine the possibility of the data being non-stationary. Given our use case, that of political party voting intentions, it is expected that new data

should be more relevant at the current time compared to older data. This is due to current events being reflected very quickly in public opinion. Treating all the data similarly as important would have the effect of a slow adaptation to sudden changes in data.

4.4.2.1 Forgetting old data

Both the word $\times$ user frequency matrices $\mathcal{Q}_i$ and the output values Y_i are likely to have means, variances and co-variances that change over time. Though non-stationary distributions can occur for a variety of reasons, one way in which they are generated is through a process similar to a random walk wherein recent values are better indicators of future values than historic values.

In this section we study the impact of downscaling weights learnt from older data. We thus attempt to deal with the potentially non-stationary nature of the training pairs, which arrive in an ordered stream. In Algorithm 1, Line 6, after a new training pair is received but before any updates are made, each of the weight vectors $\boldsymbol{u}, \boldsymbol{w}, \beta$ can be scaled down by multiplying with a value of γ, $0 \leq \gamma \leq 1$, which we name the 'forgetting factor'. The value of γ controls how quickly the old data is forgotten. When $\gamma = 1$ no modification to the online algorithm is done and if $\gamma = 0$, then only the newest input data is used. This would result in the effect of early training pairs on the model's weightings disappearing entirely as novel items are added. If the distribution of independent and dependent variables is truly non-stationary, then predictions of future variations should improve as a consequence of forgetting unhelpful old data while emphasising newer input data.

4.4.2.2 Experiments

We have performed experiments using different values for γ using the online bilinear algorithm. The model used as training and testing data the first fold of the experiments from Section 4.4.1.2.

The average RMSE as a function of γ is presented in Figure 4.16.

We can identify that the best result is obtained with a obtained with a value of $\gamma = 0.95$. Moreover, compared to the model with no forgetting factor, the model obtains better results with all values of $0.7 < \gamma < 0.98$.

This result is evidence that political voting intentions have a non-stationary distribution as the results would be expected if the distant-past was a worse predictor of the test-time period than the near-past. These results show a simple, yet effective way of improving the bilinear model's predictive ability. The appropriate amount of

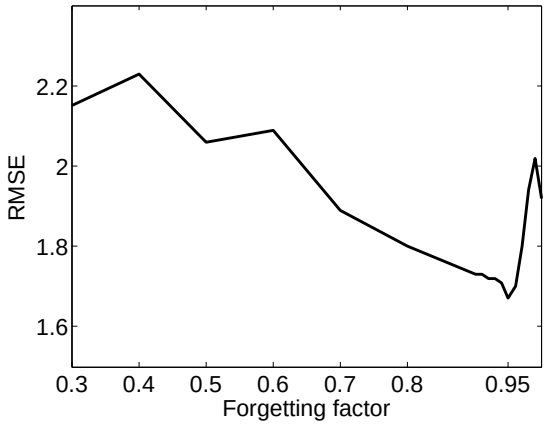


Figure 4.16: Results with varying forgetting factor.

dampening can be adjusted using validation data, as this is likely to be affected by various factors including the rate of change of underlying random walk, the rate at which data is added to the model, the optimal or desired training window.

4.5 Conclusion

In this chapter we have presented an overview that highlights the important role time has on the properties of OSN data. Analysis that look at the time series of words generated by social media data have proven useful in a number of domains and application areas. Tools, such as dimensionality reduction, further aid this kind of analysis. We have overviewed the different ways in which researchers have dealt with temporality in learning models e.g. as a variable in generative models or as regularisation in a linear model. Time also was shown to play an essential role in event detection, where word and text proprieties change in time.

Social media was shown to also be indicative of real-world activity. We have introduced models which correlate OSN data to real-world quantities (i.e. political voting intentions). These exploit both word and user spaces by solving a bilinear optimisation task and further possible relationships between the different outputs. These methods use sparse regularisers on large-scale data in order to perform noise filtering. Generality and robustness of the models were emphasised by the adaptation to two different use cases. Results have shown very good (< 1.5 RMSE) performance for political party prediction. Further, we have introduced an online learning scheme and analysed its performance, showing this is comparable to that of the batch algorithm, with the advantage of a lower computational cost for the online scheme.

We introduced a basic technique for emphasising new data, finding evidence of non-stationarity. This underscores the importance of temporal modelling.

This chapter implies that complex temporal dependencies and data volatility are important characteristics of OSN data. In the following chapters we aim to explicitly model these characteristics by building them into machine learning models and further applications. Specifically, the next chapter will deal with discovering complex temporal patterns in OSN data.

Chapter 5

Temporal prediction

Strong and complex temporal dependencies exist in OSN data as presented in the previous chapter. For modeling temporality we have incorporated into our novel models the intuition that recent data is more relevant than older data using an online learning scheme. While forgetting old data is a powerful yet simple way of modelling temporality, these methods cannot capture temporal dependencies such as periodic rise and fall. In this chapter, we aim to develop algorithms that allow modeling of more complex temporal dependencies, with an emphasis on periodicities [Preoţiuc-Pietro and Cohn, 2013a], in data from both types of OSNs under study. We consider a supervised forecasting setup where the learnt patterns are used to predict future behaviours.

We start by describing the Machine Learning framework we will use for the prediction, namely Gaussian Processes, in Section 5.1. Gaussian Processes are a Bayesian non-parametric method that is widely regarded as state-of-the-art for regression. In the following sections, we will model the OSN data using this approach for word and topics in Section 5.2 and for user behaviours in Section 5.3. We automatically identify and categorise temporal patterns and compare to a series of other competitive methods which also incorporate temporal dependencies. Conditioning on time, we aim to get a better representation of the expected events. To show this, we incorporate our forecasts into a time-aware classifier which also uses temporal effects to classify documents, rather than only words.

5.1 Gaussian Processes

In this section we present the learning framework we will use for prediction: Gaussian Process (GP) models of regression [Rasmussen and Williams, 2005]. GPs are

a probabilistic machine learning framework incorporating kernels and Bayesian non-parametrics which is widely considered as state-of-the-art for regression. The GP defines a prior over functions which applied at each input point gives a response value. Given data, we can analytically infer the posterior distribution of these functions assuming Gaussian noise. The kernel of the GP defines the covariance in response values as a function of its inputs.

We can identify two different setups for a regression problem. If the range of values to be predicted lies *within* the bounds of the training set we call the prediction task as interpolation. If the range of the prediction is *outside* the bounds, then our problem is extrapolation. In this respect, extrapolation is a more difficult task as the model needs to extract the general patterns of the data. The covariance kernel which incorporates our prior knowledge plays a major role in the prediction.

There is the case when multiple covariance kernels can describe our data. For choosing the right kernel only using the training data we employ Bayesian model selection which makes a trade-off between the fit of the training data and model complexity. We briefly overview GP regression, kernel choice and model selection. A detailed introduction to GPs is presented in Rasmussen and Williams [2005].

5.1.1 Gaussian Process regression

Consider a time series regression task where we only have one feature, the time t for which we have the value x_t. Our training data consists of n pairs $\mathcal{D} = \{(t, x_t)\}$. The model will need to predict values x_t for values of t not in the dataset using a function f learned from data.

Intuitively, the Gaussian Process defines a distribution over functions, which are subject to constraints given by the training data, i.e. passing near these values. Formally, the Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [Rasmussen and Williams, 2005]. In this case the random variables are the values of the function x_t. GP regression assumes a latent function f that is drawn from a GP prior $f(t) \sim \mathcal{GP}(m, k(t, t'))$ where m is the mean and k a kernel. The GP is completely specified by the mean m, assumed 0 in the rest of the section, and a kernel $k(t, t')$. The kernel specifies the covariance between pairs of outputs:

$$k(t, t') = \text{cov}(x_t, x'_t) \tag{5.1}$$

Given the covariance is defined over an infinite set of pairs, we need to assume a simple form of the covariance by defining covariance values using a kernel function.

The covariance implies a distribution over functions and encodes the properties of the functions (e.g. smoothness, periodicity, stationarity). We illustrate this by generating random functions $f \sim \mathcal{N}(0, k(t, t'))$ where k is a squared exponential kernel that defines smooth functions (defined in Section 5.1.2) and t are equally spaced data points. Different samples of functions are presented in Figure 5.1.

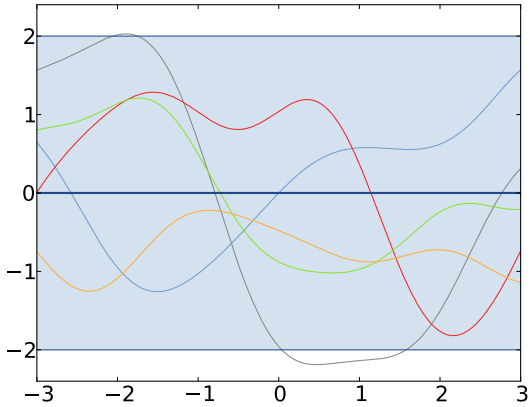


Figure 5.1: Functions drawn from a GP prior with a standard smooth covariance (Squared Exponential, Section 5.1.2). The thick line represents the mean and the shaded area represents the 95% confidence region.

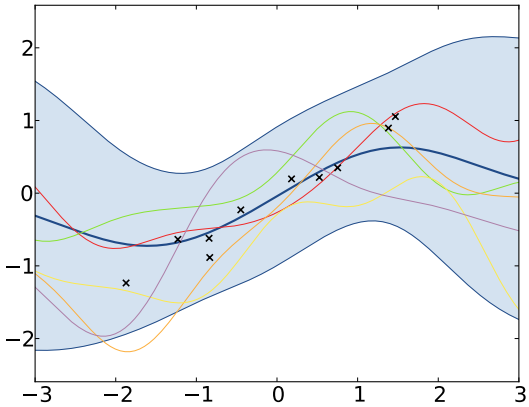


Figure 5.2: Functions drawn from the posterior after observing 10 noisy samples (marked with crosses). The thick blue line represents the mean and the shaded area represents the 95% confidence region.

We assume the predicted value is obtained by the function evaluated at the corresponding data point, $x_t = f(t) + \eta_t$, where $\eta \sim \mathcal{N}(0, \sigma^2)$ is white-noise. In order to make predictions incorporating our belief over possible functions after observing the training set $\mathcal{D}$ we must compute the posterior distribution over functions. For this, we need to restrict the joint prior distribution of the training and test points to generate the observed training points. The posterior at a test point t_* is given by:

$$p(x_*|t_*, \mathcal{D}) = \int_f p(x_*|t_*, f) \cdot p(f|\mathcal{D}) \tag{5.2}$$

where x_* and t_* are the test response and input. The predictive posterior can be solved analytically with the solution:

$$x_* \sim \mathcal{N}(k_*^T (K + \sigma_n^2 I)^{-1} \mathbf{t}, k(t_*, t_*) - k_*^T (K + \sigma_n^2 I)^{-1} k_*) \tag{5.3}$$

where $k_* = [k(t_*, t_1) ... k(t_*, t_n)]^T$ are the kernel evaluations between the test point and all the training points, $K = \{k(t_i, t_j)\}_{j=1..n}^{i=1..n}$ is the Gram matrix over the training

points and $\mathbf{t}$ is the vector of training points. The posterior of x_* includes the mean response as well as its variance, thus expressing the uncertainty of the prediction. The forecast is usually taken as the expected value. Sample functions taken from the posterior after observing a few data points are presented in Figure 5.2. We note that due to the matrix inversion in 5.3, exact inference is of complexity $\mathcal{O}(n^3)$, where n is the number of training points. Efforts to scale GPs to a large number of variables are well understood [Candela and Rasmussen, 2005] and approximate a solution in complexity to $\mathcal{O}(nm^2)$, where m is selected at runtime, with a higher value giving better approximations.

5.1.2 Kernels

The kernel together with its parameters fully define the GP. The kernel induces the covariance in the response between pairs of data points. Intuitively, if the desired function is smooth, closer points should have high covariance compared to points that are further apart. If a periodic behaviour is desired, points at period p distance should have the highest covariance. Usually, the kernel is isotropic i.e. invariant to all rigid motions.

For interpolation, a squared exponential kernel is usually used which encourages smooth functions. Figure 5.3 shows regression over three days for the frequency of the hashtag #goodmorning on Twitter when only a random third of the values of the function are observed. We see that both a smooth kernel (Squared Exponential, defined below) and a periodic kernel (Periodic Spikes, defined below) give good results. This is because in interpolation scenarios a smoothly transitioning function between observations usually provides a good fit. We notice however that in intervals when no observations are provided (as in the extrapolation scenario) the variance of the prediction is very high.

Consequently, for extrapolation, the choice of the kernel is paramount. The kernel encodes the prior belief about the type of function we aim to learn. To illustrate this, in Figure 5.4, we show the same time series over a larger 2 weeks interval and plot the regression for the future week learned by using different kernels.

We will use multiple kernels, each most suitable for a specific category of temporal patterns in our data. This includes a new kernel inspired by observed word occurrence patterns. The kernels are:

Constant (Const): The constant kernel is $k_C(t, t') = c$ and it describes a constant relationship between outputs. Its mean prediction will always be the value c learned in training. Its assumption is that the signal is modeled only by Gaussian

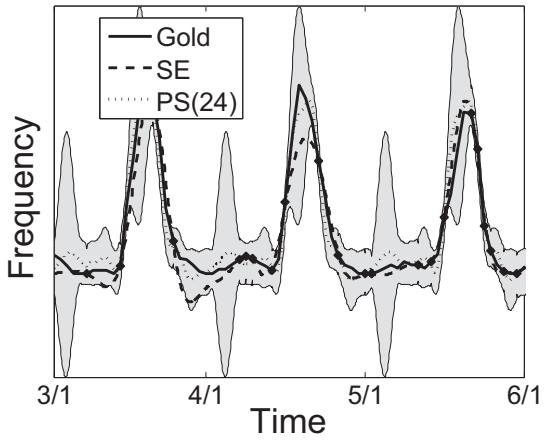


Figure 5.3: Interpolation for #goodmorning over 3 days with SE and PS(p=24,s=3) kernels. Prediction variance shown in grey for PS(24). Crosses represent training points.

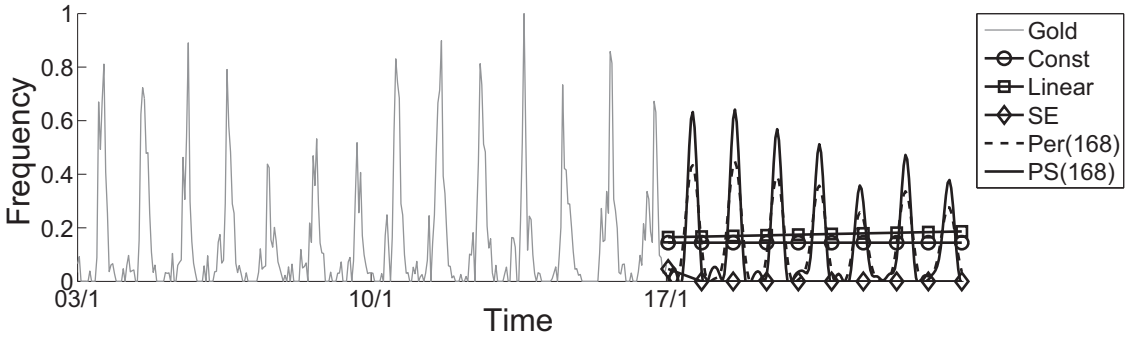


Figure 5.4: Extrapolation for #goodmorning over 3 weeks with GPs using different kernels. Forecast is the predictive mean.

noise centred around this value. This describes the data best when we have a noisy signal around a stationary mean value.

Squared exponential (SE): The SE kernel or the Radial Basis Function (RBF) is the standard kernel used in most interpolation settings:

$$k_{SE}(t,t') = s^2 \cdot \exp\left(-\frac{(t-t')^2}{2l^2}\right) \tag{5.4}$$

This gives a smooth transition between neighbouring points and best describes time series with a smooth shape e.g. a uni-modal burst with a steady decrease. However, the predictive variance increases exponentially with distance. Predictions well into the future will have no covariance. Its two parameters s and l are the characteristic length-scales along the two axes. Intuitively, they control the distance of inputs on a particular axis from which the function values become uncorrelated. We note that

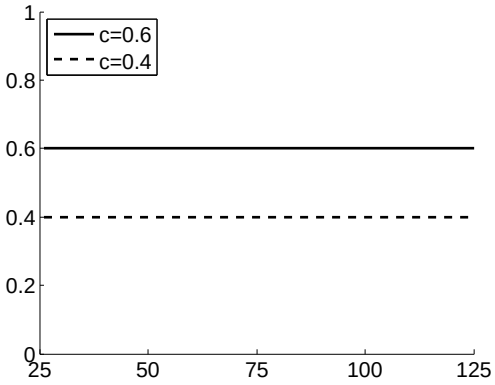


Figure 5.5: Constant kernel.

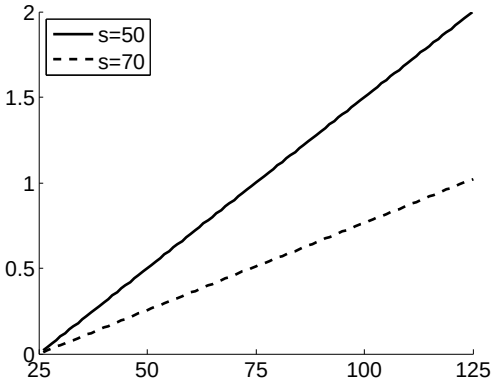


Figure 5.6: Linear kernel.

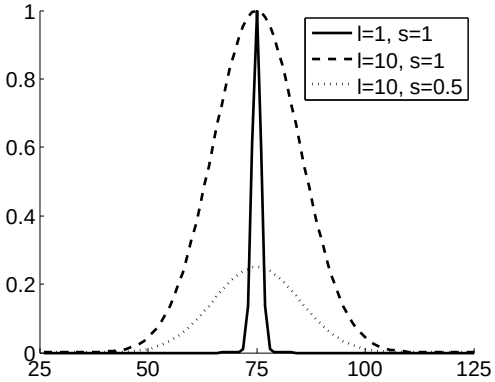


Figure 5.7: Squared exponential kernel.

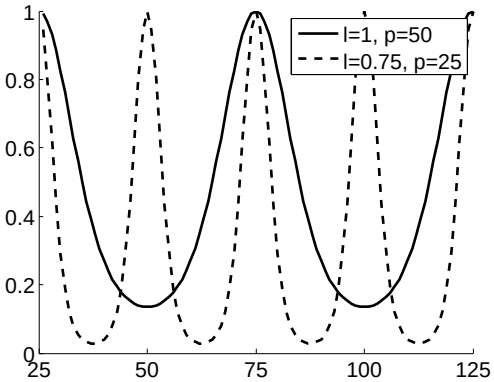


Figure 5.8: Periodic kernel.

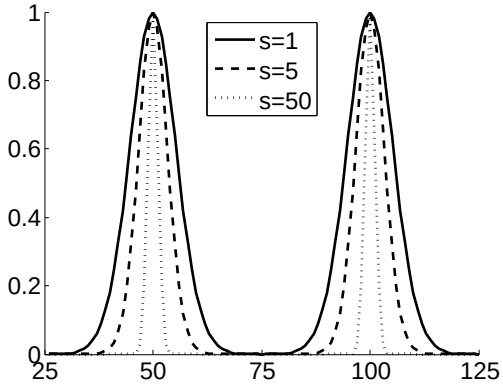


Figure 5.9: Behaviour of the PS kernel with p=50.

using the SE kernel in a Gaussian Process corresponds to Bayesian linear regression with an infinite number of basis functions [Rasmussen and Williams, 2005].

Linear (Lin): The linear kernel describes a linear relationship between outputs:

$$k_{Lin}(t, t') = \frac{|t \cdot t'| + 1}{s^2} \tag{5.5}$$

This kernel is non-stationary as the covariance between outputs does not depend only on the difference between data points $|t - t'|$, but on their actual values. Gaussian processes using the linear kernel is equivalent to Bayesian linear regression with $\mathcal{N}(0, 1)$ priors on the corresponding regression weights and a prior of $\mathcal{N}(0, s^2)$ on the bias.

Periodic (PER): The periodic kernel represents a SE kernel in polar coordinates and describes a sinusoidal relationship between outputs:

$$k_{PER}(t, t') = s^2 \cdot \exp \cdot \left(-\frac{2\sin^2(2\pi(t - t')/p)}{l^2} \right) \tag{5.6}$$

The kernel is good at modelling periodically patterns that oscillate smoothly between low and high frequency. s and l are characteristic length-scales as in the SE kernel and p is the period i.e. distance between consecutive peaks.

Periodic spikes (PS): We introduce this kernel in order to model the following periodic behaviour: abrupt periods of high values, usually with a peak, followed by periods of very low occurrence:

$$k_{PS}(t, t') = \cos\left(\sin\left(\frac{2\pi \cdot (t - t')}{p}\right)\right) \cdot \exp\left(\frac{s\cos(2\pi \cdot (t - t'))}{p} - s\right) \tag{5.7}$$

This is inspired by observations on studying word frequencies for which low values can be observed for short or long time intervals, followed by abrupt periodic rise in usage. For example, words associated with a weekly TV series will only have non-zero frequency during and around its air time. Some other words will only be used at close to nighttime and seldom during the rest of the day.

The kernel is parametrised by its period p and a shape parameter s. The period indicates the time interval between the peaks of the function, while the shape parameter controls the *width* of the spike. The behaviour of the kernel is illustrated in Figure 5.9. We constrain $s \geq 1$.

In Figure 5.4 we illustrate how the forecast is highly dependent on the kernel choice. When having as input this periodic time series, the PER and PS kernels will normally model and forecast best. The PS kernel works better because it can better model the sharp peaks whilst also fitting the abrupt drop and very small values. The PER kernel underpredicts the peaks and only allows for a smoother drop in usage. We use for both kernels a period of 168 hours. This is because although a daily pattern exists, the weekly one is stronger, with the day of the week influencing the volume of the hashtag. NRMSE (Normalised Root Mean Square Error) in Table 5.1 on the held out data confirms this finding, with PS showing the lowest error. Examples when each

other kernel perform best are presented further in Figure 5.11. The flexibility of the GP framework also allows us to combine kernels (e.g. SE · PS or PS + Lin) in order to identify a combination of trends [Duvenaud et al., 2013, Gönen and Alpaydin, 2011]. Experiments on subsets of data showed no major benefits of combining kernels, but the computational time and model complexity increased drastically due to the extra hyperparameters.

	Const	**Lin**	**SE**	**PER**	**PS**
NLML	-41	-34	-176	-180	-192
NRMSE	0.213	0.214	0.262	0.119	0.107

Table 5.1: Negative Log Marginal Likelihood (NLML) shows the best fitted model for the #goodmorning time series in Figure 5.4. NRMSE computed on the third unobserved week. Lower values are better in both cases.

5.1.3 Model selection and optimisation

We briefly overview the concepts of model selection in the GP framework. By this we refer to multiple aspects such as choosing the model (kernel) from a discrete set $\mathcal{H} = \{H_i\}_{i=1}^{n}$ and optimising the kernel parameters θ (which we refer to as the model hyperparameters). In the GP Bayesian inference scheme, we can compute the probability of the data given the hyperparameters and model marginalising over the function space. This is called the *marginal likelihood* or *evidence* and is useful for model selection using only the training set:

$$p(\boldsymbol{x}|\mathcal{D}, \theta, H_i) = \int_f p(\boldsymbol{x}|\mathcal{D}, f, H_i)p(f|\theta, H_i) \tag{5.8}$$

where $\boldsymbol{x}$ is the vector of outputs. For finding the hyperparameters θ, instead of computing the posterior using Bayes rule:

$$p(\theta|\boldsymbol{x}, \mathcal{D}, H_i) = \frac{p(\boldsymbol{x}|\mathcal{D}, \theta, H_i)p(\theta|H_i)}{p(\boldsymbol{x}|\mathcal{D}, H_i)} \tag{5.9}$$

we approximate it using a point estimate by minimising the negative log of Equation 5.8 over hyperparameters θ. This approximation is also known as type II maximum likelihood (ML-II). The negative log of the evidence (Equation 5.8 i.e. Negative Log Marginal Likelihood or NLML) for GP regression can be computed analytically:

$$-\log p(\boldsymbol{x}|\mathcal{D}, \theta, H_i) = \frac{1}{2}\boldsymbol{x}K_{\boldsymbol{x}}^{-1}\boldsymbol{x} + \frac{1}{2}\log|K_{\boldsymbol{x}}| + \frac{n}{2}\log 2\pi \tag{5.10}$$

where $K_{\boldsymbol{x}} = K_f + \eta_n^2\mathbb{I}$ and K_f is the covariance matrix for the latent function f.

The NLML balances data fit and the model complexity by automatically incorporating the Occam's Razor principle namely that 'the simplest solution is to be preferred over a more complex one') [Rasmussen and Ghahramani, 2000]. By examining the components of the NLML, $\frac{1}{2}\boldsymbol{x}K_{\boldsymbol{x}}^{-1}\boldsymbol{x}$ represents the data fit of the observed values, while $\frac{1}{2}\log|K_{\boldsymbol{x}}|$ represents a complexity penalty which depends only on the kernel and its hyperparameters. The final term is a normalisation constant.

Our second goal is to identify the best model H_i (kernel) from a set $\mathcal{H}$. The posterior for a model is:

$$p(H_i|\boldsymbol{x},\mathcal{D}) = \frac{p(\boldsymbol{x}|\mathcal{D},H_i)p(H_i)}{p(\boldsymbol{x}|\mathcal{D})} \tag{5.11}$$

Assuming a uniform prior over the models in $p(\theta|H_i)$, the posterior for a model is then proportional to:

$$p(H_i|\boldsymbol{x},\mathcal{D}) \propto p(\boldsymbol{x}|\mathcal{D},H_i) = \int_\theta p(\boldsymbol{x}|\mathcal{D},\theta,H_i)p(\theta|H_i) \tag{5.12}$$

The marginal likelihood $p(\boldsymbol{x}|\mathcal{D},H_i)$ is the probability of the data given the model and is approximated by taking the maximum. The evidence has to normalise and this way, complex models which can account for many datasets achieve low Bayesian evidence. One can think of the evidence as the probability that a random draw of a function from the model class would generate the dataset $\mathcal{D}$. Complex models are penalised because they can describe many datasets, while the simple models can describe only a few datasets, thus the chance of a good data fit is very low.

This is for example the case of the periodic bursts in Figure 5.4. Although the periodic kernel can fit the data, it will incur a high model complexity penalty. Introducing a periodic covariance between outputs at p intervals using the PER kernel also includes multiple covariances which are not necessary in describing the data. This results in a high rank/determinant of $K_{\boldsymbol{x}}$ in Equation 5.10. Intuitively, the resulting model can describe many more functions than the one observed. The PS kernel introduces less covariance other than between outputs at p intervals resulting in a sparser $K_{\boldsymbol{x}}$ and a lower complexity penalty. Consequently the model can describe fewer functions while still fitting the data. The PS kernel in this respect is a simpler model and the preferred one.

When the dataset is observed, the evidence can select between the models. More generally, the model choice actually gives an *implicit classification* of the temporal patterns depending on the kernel choice we make. We can thus identify which data is best fit by a non-stationary kernel (e.g. linear, polynomial or neural network [Neal,

1996]) as opposed to a stationary one. By considering the stationary kernels from the previous section and a one-dimensional input we can identify the following classes: a steady signal with noise (**Const** kernel), a signal with local temporal patterns (**SE** kernel), an oscillating periodic pattern (**PER** kernel) or a pattern with abrupt periodic peaks (**PS** kernel).

In our experiments, we use the NLML for optimising the hyperparameters only using training data. For optimising the parameters of the kernel introduced in Equation 5.7, it is important to first identify the right period. We consider as possible periods all integer values less than half the size of the training set, and then tune the shape parameter using gradient descent to minimise NLML. We then take the *argmin* value of those considered. We show the NLML for a sample regression in Figure 5.10.

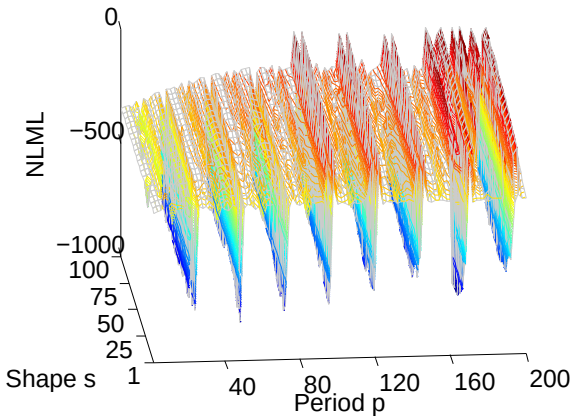


Figure 5.10: NLML for #goodmorning on the training set as a function of the 2 kernel parameters.

The figure shows that there are multiple canyons in the likelihood, which can lead a convex optimisation method to local optima. These appear when p is equal or an integer multiple of the main period of the data, in this case 24. The lowest values are obtained when $p = 168$, allowing the model to accommodate the day of week effect. Our procedure is not guaranteed to reach a global optima, but is a standard technique for fitting periodic kernels [Duvenaud et al., 2013].

5.2 Temporal modelling of words

Text use changes with time as we have seen in Chapter 4 and can affect many downstream applications. In this section we mainly model periodic distributions of words over time, which have largely been ignored in NLP applications. Our use case is

hashtag frequency in Twitter and first automatically identify the periodic patterns and their shape. We propose a model that first broadly identifies several types of temporal patterns: a) periodic, b) constant in time, c) falling out of use after enjoying a brief spell of popularity (e.g. Internet memes, news). This is performed automatically only using training data and makes no assumptions on the existence or the length of the periods we aim to model. The model is then used to forecast the frequency of a hashtag.

We expect text usage to follow multiple periodicities at different scales. For example, people on social media might talk about different topics during and after work on weekdays, talk every Friday about the weekend ahead, or comment about their favorite weekly TV show during its air time. Given this, text frequencies will display periodic patterns. This applies to other text related quantities like co-occurrence values or topic distributions over time, as well as applications like user behaviour, which we consider in the next section.

Modelling temporal patterns and periodicities can be useful to downstream tasks like text classification. We use the GP forecast for a hashtag's frequency in a time interval as a prior in a Naïve Bayes model for text classification and compare to methods which do not account for temporal periodicities. For example, a tweet containing 'music' is normally attributed to a general hashtag about music like #np (now playing). However, we can exploit information about the time the tweet was authored. If this is during the (weekly periodic) air time of 'American Idol' it is more likely for it to belong to #americanidol or if its mentioned in the days building up to the Video Music Awards to be assigned to #VMA. The predicted values are thus incorporated in the classification task of assigning the tweet hashtag based on the rest of its text.

5.2.1 Task

We treat our task of forecasting the volume of a Twitter hashtag as a regression problem. The hashtag volume is aggregated into one hour time intervals. Because the total number of tweets varies depending on the day and hour of day, we chose to model the proportion of tweets with the given tag in that hour. Given a time series of these values as the training set for a hashtag, we aim to predict the values in the testing set, extrapolating to the subsequent month.

Hashtags represent free-form text labels that authors add to a tweet in order to enable other users to search them to participate in a conversation defined by these (see also Section 2.1.1). They group similar tweets (like topics), reflect real world

events (some of which are periodic) and present direct means of evaluation. Note that this approach could be applied to many other temporal problems in NLP or other domains. We treat each regression problem independently, learning for each hashtag its specific model and set of parameters. Although multi-task learning [Álvarez et al., 2012] can be applied to share hyperparameters across regression tasks, this incurs a large increase in processing time and preliminary experiments showed no major boost in performance.

5.2.2 Related work

Modelling periodicities is one of the standard applications of Gaussian Processes [Rasmussen and Williams, 2005]. Recent work by Wilson and Adams [2013] and Durrande et al. [2013] show how different periods can be identified from data. Methods that assume certain periodicities at daily or weekly levels were proposed in [McInerney et al., 2013]. With text application, GPs were used by Polajnar et al. [2011] for protein interaction detection in biomedical texts, by Cohn and Specia [2013] for modelling subjective human evaluations of machine translation quality, by Lampos et al. [2014] to predict user impact on Twitter based on text and profile features and by Specia et al. [2013] and Shah et al. [2013] in feature selection for predicting machine translation quality.

For predicting future popularity of hashtags, Tsur and Rappoport [2012] use linear regression with a wide range of features. Ma et al. [2012, 2013] frame the problem as classification into a number of pre-determined frequency intervals and apply all the standard classifiers. None of these studies model periodicities, although the former stresses their importance for accurate predictions. For predicting the hashtag given the tweet text, Mazzia and Juett [2011] use the Naïve Bayes classifier with the uniform and empirical prior or TF-IDF weighting. Zangerle et al. [2011] use cosine similarity between tweets to recommend hashtags at write time while Godin et al. [2013] uses topic models to identify the most likely hashtag for a tweet. Hashtag recommendation using content-based collaborative filtering was explored in [Kywe et al., 2012]. Our model incorporates in addition information about temporal periodicities.

5.2.3 Experimental setup

For our experiments we used Twitter data from the Gardenhose dataset (described in Section 2.1.3) in the interval 1 January – 28 February 2011. For simplicity in the classification task, we filtered the stream to include only tweets that have exactly one

hashtag. These represent approximately 7.8% of our stream. As text processing steps, we have tokenised all the tweets, filtered them for English and removed duplicate tweets using methods described in Chapter 3.

In our experiments we use the first month of data as training and the second month as testing. Note the challenging nature of this testing configuration where predictions must be made for up to 28 days into the future. We keep a total 1176 of hashtags which appear at least 500 times in each split of the data. This is done in order filter the many hashtags which are short lived and are not interesting for long term forecasting. The average frequencies of the hashtags is 5456. The vocabulary consists of all the tokens that occur more than 100 times in the dataset and start with an alphabetic letter. After processing, our dataset consists of 6,416,591 tweets with each having on average 9.55 tokens and each hashtag appearing on average in 5456 tweets.

5.2.4 Methods

We choose multiple baselines for our prediction task in order to compare the effectiveness of our approach. These are:

Mean value (M): We use as prediction the mean of the values in the training set. Note that this is the same as using a GP model with a constant kernel (+ noise) with a mean equal to the training set mean.

Lag model with GP determined period (Lag+): The prediction is the mean value in the training set of the values at lag δ where δ is the period rounded to the closest integer as determined by our GP model. This is similar to an auto-regressive (AR) model with all the coefficients except δ set to 0. Please note that given the period δ this is a very strong model as it gives a mean estimate at each point. Comparing to this approach we can see if the GP model can recover the underlying function that described the periodic variation and filter out the noise in the observations. Correctly identifying the period is very challenging as we discuss in Section 5.2.6.

GP regression: Gaussian Process regression using either the SE kernel (**GP-SE**), the periodic kernel (**GP-PER**), the PS kernel (**GP-PS**). The method that chooses between kernels using model selection as described in Section 5.1.3 is denoted as **GP+**. We will also compare to GP regression the linear kernel **(GP-Lin)**, but we will not use this as a candidate for model selection due the poor results shown below. Experiments on a subset of data showed no major benefits of combining kernels, but the computational time and model complexity increased drastically due to the extra hyperparameters. However, in our experimental setup where we use a relatively

small time frame, there are few linear trends in the data. It might seem limiting that we only learn a single period, although we could combine periodic kernels with different periods together. But, as we have seen in the #goodmorning example (with overlapping weekly and daily patterns), if there is a combination of periods the model can select a single period which is the least common multiple.

5.2.5 Results

We start by qualitatively analysing a few sample regressions that are representative of each category of time series under study. These are shown in Figure 5.11. For clarity, we only plotted a few kernels on each figure. The full evaluation statistics in Normalised Root Mean Squared Error (NRMSE) computed on the test set and the Bayesian evidence on the training set are show in Table 5.2.

Hashtag	**Lag+**	**Const**		**SE**		**PER**		**PS**	
	NRMSE	NLML	NRMSE	NLML	NRMSE	NLML	NRMSE	NLML	NRMSE
#fyi	0.1578	**-322**	**0.1404**	-320	0.1898	-321	0.1405	-293	0.1456
#confessionhour	0.0404	-85	0.0107	**-186**	**0.0012**	-90	0.0327	-88	0.0440
#fail	0.1431	-376	0.1473	-395	0.4695	**-444**	**0.1387**	-424	0.1390
#breakfast	0.1363	-293	0.1508	-333	0.1773	-293	0.1514	**-367**	**0.1276**
#raw	0.0464	-1208	0.0863	-1208	0.0863	-1323	0.0668	**-1412**	**0.0454**

Table 5.2: NRMSE shows the best performance for forecasting and NLML (only using training data) shows the best model for all the regressions in Figure 5.11. Lower is better.

For the hashtag #fyi there is no clear pattern. For this reason the model that uses the constant kernel performs best, being the simplest one that can describe the data, although the others give similar results in terms of NRMSE on the held-out testing set. While time series modelled using this kernel do not significantly outperform all others on NRMSE on held-out data, the kernel is useful for interpreting the time series, separating noisy ones from those having an underlying periodic behaviour.

The #confessionhour (the hashtag tags tweets of people posting things they have never told anyone) example illustrates a behaviour best suited for modelling using the SE kernel. We notice a sudden burst in volume which decays over the next 2 days. This is actually the behaviour typical of 'Internet memes' as presented in Yang and Leskovec [2011]. These cannot be modelled with a constant or periodic kernel as shown by the results on held-out data and the time series plot. The periodic kernels will fail in trying to match the large burst with others in the training data and will attribute to noise the lack of a similar peak, thus making bad predictions. In this

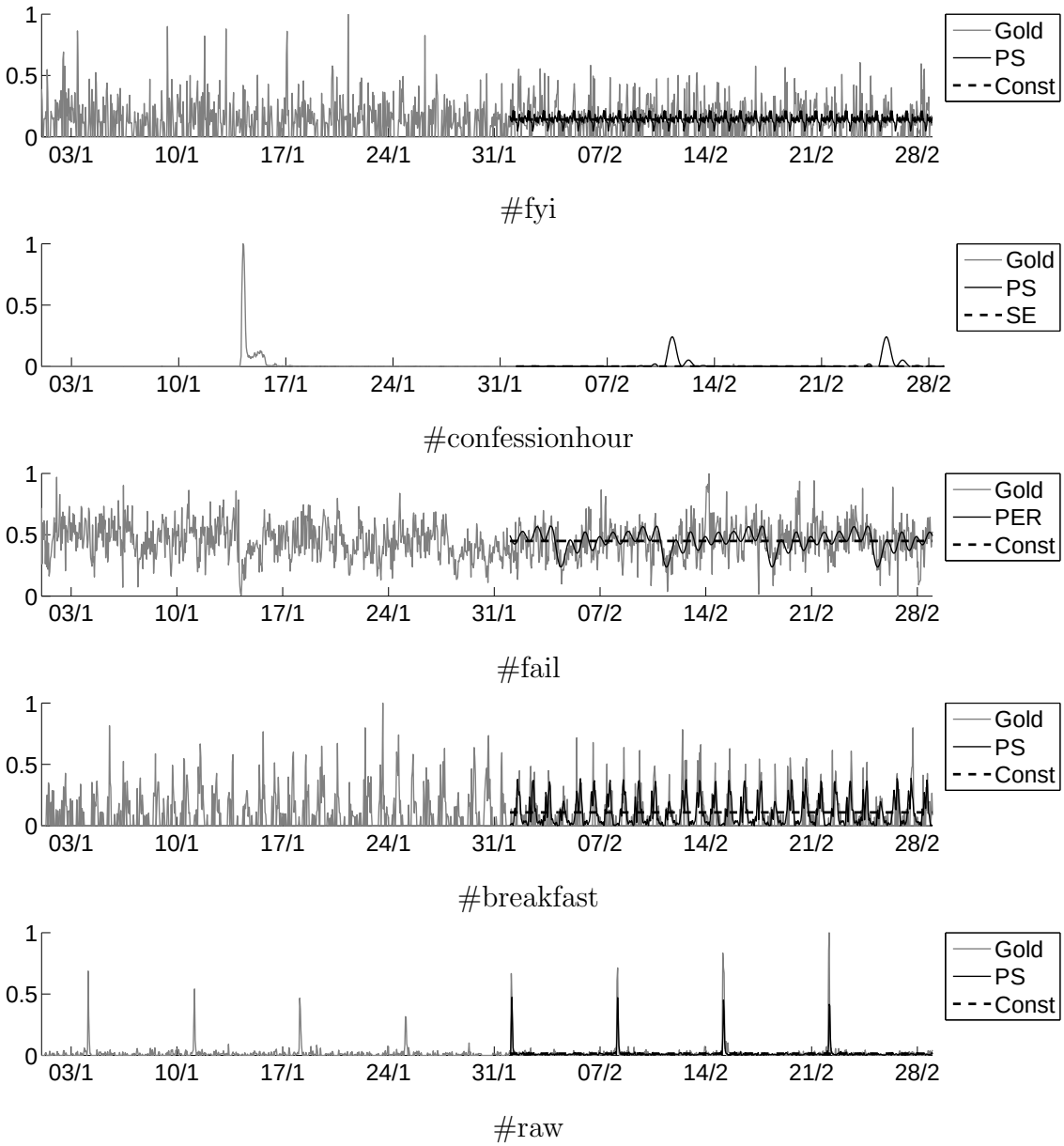


Figure 5.11: Sample regressions and their fit using different methods.

example, forecasts will be very close to 0 under the SE kernel, which is what we would desire from the model.

The periodic kernel best models hashtags that exhibit an oscillating pattern. For example, this best fits words that are used frequently during the day and less so during the night, like #fail (used by people tweeting about their or others' failures). Here, the period is chosen to be one week (168) rather than one day (24) matching the weekly effect superimposed on the daily one. Our model recovers that there is a daily pattern with a higher frequency during the day. On weekends however, and

especially on Friday evenings, people talk are more preoccupied by other things.

The PS kernel introduced in this chapter models best hashtags that have a large and short lived burst in usage. We illustrate this by two examples. First, we choose #breakfast which has a daily and weekly pattern. As we would expect, a big rise in usage occurs during the early hours of the day, with very few occurrences at other times. Our model discovers a weekly pattern as well. This is used mainly for modelling the difference between weekends and weekdays. On weekends, the breakfast tag is more evenly spread during the hours of the morning, because people who do not have to wake up for work can have breakfast at a more flexible time than during the week. In the second example, we present a hashtag that is associated to a weekly event: #raw is used to discuss a wrestling show that airs every week for 2 hours on Monday evenings in the U.S.. With the exception of these 2 hours and the hour preceding it, the hashtag is rarely used. This pattern is modelled very well using our kernel, with a very high value for the shape parameter ($s = 200$) compared to the previous example ($s = 11$) which captures the abrupt trend in usage. In all cases, our GP model chosen by the evidence performs better than the Lag+ model, which is a very strong method if presented with the correct period. This further demonstrates the power of the Gaussian Process framework to deal with noise in the training data and to find the underlying function of the time variation of words.

In Table 5.3 we present sample tags identified as being part of the four hashtag categories, and the total number of hashtags in each. We notice a significant number of bursty hashtags identified by the **SE** kernel. These are either 'Internet memes' (e.g. #backintheday, #questionsidontlike) which were prevalent on the training set and decayed in frequency in the test set, while still being mentioned. Another type of hashtags in this class is the ones that mention events which peaked in January (e.g. #haiti or #snowday). Because we keep only hashtags that occurred above the threshold (of 500 in each split) in both the training and testing sets, we discard the vast majority ($\sim 90\%$) of hashtags which only occur for less then one day usually and which normally would belong to this class. The **Const** kernel is selected for the lowest number of hashtags. This is both caused by both the day/night effect which makes many hashtags periodic and the bursty effect described by the SE kernel. Hashtags include those used constantly by Twitter users (e.g. #nf – now following, #funny). The **PER** kernel identifies hashtags which mainly exhibit a day/night effect, such as #coffee or #xbox, but without a rapid decay at any point in time. Conversely, the **PS** kernel identifies hashtags that suffer from a high decay, being hardly used at some periods of time. For example, #tgif ('Thank God is Friday'), #ff ('Follow

Friday'), #ww ('Writers Wednesday') only occur on one day of the week. Similarly, #goodnight and #twitteroff are frequent only in the evenings when users take a break from posting on their Twitter account, while #jobs is frequent only during working hours as people use the hashtag to announce job opportunities.

Const	**SE**	**PER**	**PS**
#funny	#2011	#brb	#ff
#lego	#backintheday	#coffee	#followfriday
#likeaboss	#confessionhour	#facebook	#goodnight
#money	#februarywish	#facepalm	#jobs
#nbd	#haiti	#fail	#news
#nf	#makeachange	#love	#nowplaying
#notetoself	#questionsidontlike	#rock	#tgif
#priorities	#savelibraries	#running	#twitterafterdark
#social	#snow	#xbox	#twitteroff
#true	#snowday	#youtube	#ww
49	**268**	**493**	**366**

Table 5.3: Sample hashtags for each category. The last line shows the total number of hashtags of each type.

As a means of quantitative evaluation we compute the relative NRMSE on the test set compared to the Mean (M) method for forecasting. We choose this, because we consider that NRMSE is not comparable between regression tasks due to the presence of large peaks in many time series, which distort the NRMSE values. The results are presented in Table 5.4 and show that Gaussian Processes with model selection performs best. Remarkably, it consistently outperforms the Lag+ method, which shows the effectiveness of the GP models to incorporate uncertainty and to capture significant periodic patterns. The GP-PS model does very well on its own. Although chosen in the model selection phase in only a third of the tasks, it performs consistently well across tasks because of its ability to model well all the periodic hashtags, be they smooth or abrupt. The GP-Lin model does worse than the average, mostly due to time series with a few modes for which the GP-Lin model will spread the mass across time.

Lag+	**GP-Lin**	**GP-SE**	**GP-PER**	**GP-PS**	**GP+**
7.29%	-3.99%	-34.5%	0.22%	7.37%	**9.22%**

Table 5.4: Average relative gain over mean (M) prediction for forecasting on the entire month.

5.2.6 Discussion

Let us now turn to why the GP model is better for discovering periodicities than some classic time series modelling methods. Measuring auto-correlation between points in the time series is used to discover the hidden periodicities in the data and in building AR models. The auto-correlation computes the similarity between points in the time series as a function of the time lag. The period candidates are represented by the time lag value corresponding to peaks in the auto-correlation function. The first problem of this method is that if a period exists, all integer multiples will be feasible candidates. Another problem is that of auto-correlated noise e.g. occurring in a burst over a time window. This case is illustrated in Figure 5.12 where #confessionhour shows auto-correlation at low time lag values with a peak. However, as seen in Figure 5.11, the time series lacks a periodic component.

Another approach to discovering periods in data is by computing the power spectral density. The spectrum decomposes a time series (or stochastic process) into the frequencies present in that process. The periods are indicated by peaks in the periodogram at the respective frequency value. This works best in case of time series that have a short Fourier series decomposition which expresses the time series as a sum of oscillating patterns. However, this is incapable to clearly discover the correct period when dealing with time series with large periodic bursts like those exhibited by the #raw time series as shown in Figure 5.13. The Fourier decomposition cannot model step-functions and other non-smoothly varying signals. In Figure 5.13, the lowest frequency spike corresponds to the correct period of 168, but also other candidate periods are shown as almost equally likely. This method has been used in the GP framework before by Wilson and Adams [2013] for initialising the periodic component of a periodic kernel.

5.2.7 Text based prediction

In this section we demonstrate the usefulness of our model in a downstream task: predicting the hashtag of a tweet based on its text. In contrast to this classification approach for suggesting a tweet's hashtag, information retrieval methods based on computing similarities between tweets are hard to scale to large data [Zangerle et al., 2011].

We choose a simple model for prediction, the Naïve Bayes Classifier, which provides us with a straightforward way to incorporate our prior knowledge of how frequent a hashtag h_k is in a certain time frame. The Naïve Bayes Classifier chooses for

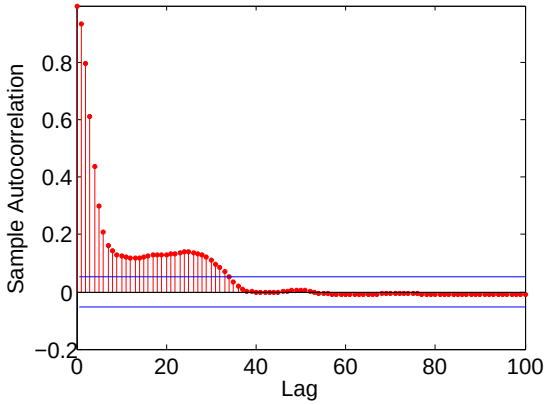


Figure 5.12: Sample auto-correlation for #confessionhour.

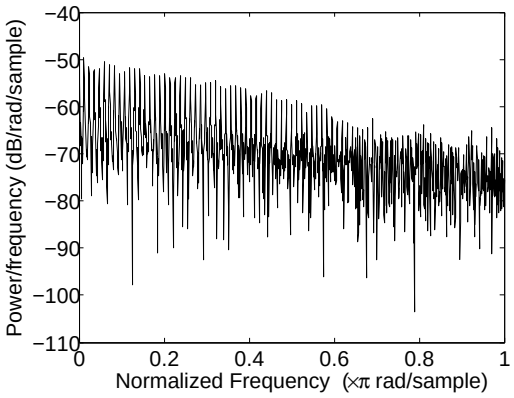


Figure 5.13: Periodogram Power spectral density estimate for #raw.

a tweet with the bag of words W, the hashtag h^*:

$$h^* = \arg\max_{k=1,\ldots,K} \left(\log p\left(h_k\right) + \log p\left(W \mid h_k\right)\right) \tag{5.13}$$

$$\log p\left(W \mid h_k\right) = \sum_{w_j \in W} p\left(w_j \mid h_k\right) = \sum_{w_j \in W} \frac{\text{freq}(w_j, h_k)}{\text{freq}(w_j)} \tag{5.14}$$

where by $\text{freq}(w_j)$ is the frequency of word w_j in the training set and $\text{freq}(w_j, h_k)$ is the frequency of word w_j appearing in tweets with h_k hashtag. $p\left(h_k\right)$ is the prior probability of a tweet having hashtag h_k. The empirical prior (**NB-E**) estimates the prior from training data: $\log p\left(h_k\right) = \text{freq}(h_k)$.

We use as prior in the Naïve Bayes model (**NB-P**) the forecast for the hour (t) the target tweet was authored in: $\log p\left(h_k\right) = h_{kt}$. In our comparison we use the Most Frequent (**MF**) baseline. Because there are more than 1000 possible classes we show the accuracy of the correct hashtag being amongst the top 1,5 or 50 hashtags as well as the Mean Reciprocal Rank (MRR). The results are shown in Table 5.5.

	MF	**NB-E**	**NB-P**
Match@1	7.28%	16.04%	**17.39%**
Match@5	19.90%	29.51%	**31.91%**
Match@50	44.92%	59.17%	**60.85%**
MRR	0.144	0.237	**0.252**

Table 5.5: Results for hashtag classification. Higher is better.

The results show that incorporating the forecasted values as a more informative prior for classification we obtain better predictions. The improvements are consistent in all the Match values. Also, we highlight that a 9% improvement in the forecasting

Tweet	Time	Prior	Rank	Prediction
Bruins Goal!!! Patrice Bergeron makes it 3-1 Boston	2-3am 2 Feb 2011	E: 0.00017 P: 0.00086	7 1	#fb #bruins
i need some of Malik people	3-4am 2 Feb 2011	E: 0.00021 P: 0.00420	7 1	#ff #thegame
Alfie u doughnut! U didn't confront Kay? SMH	7-8pm 3 Feb 2011	E: 0.00027 P: 0.00360	8 1	#nowplaying #eastenders

Table 5.6: Example of tweet classification using the Naïve Bayes model with the two different priors (E - empirical, P - GP forecast). Rank shows the rank in probability of the correct class (hashtag) under the model. Time is G.M.T.

task carries over to about a 2% improvement in classification. Moreover, treating the entire classification task in a GP modelling framework can lead to further improvements, but is subject to scaling issues for a large number of candidate classes. In Table 5.6 we show a few examples in which the GP learned prior makes a difference in classification together with the values for both priors.

5.3 Temporal modelling of user behaviour

Time plays a very important factor in user behaviour as presented in Section 2.2.2.1. In this section we use LBSN data where a data point represent a user checking-in to a specific location, called venue. Temporal information, such as the local time or the day of the week influences the user's choice of location. We aim to use this observation by adding periodicity information to the task of next location prediction. Because there is a very high potential number of places a user will visit, some of which might not have been observed before, the prediction task will suffer from data sparsity. For alleviating this problem, we use as an intermediate representation of the location, i.e. its category. The categories are a discrete set of nine different types, each with its own characteristics.

We study the temporal patterns of venue categories in Section 5.3.1. Observing such patterns can aid understanding human mobility with an immediate application in predicting future user location. For this task, simple Markov Models have proven accurate in previous work. Consequently, we also perform a study of transitions between venue categories, showing which transition is prevalent given the current venue. The underlying data is represented by the 'Frequent Users Dataset' introduced in Section 2.2.2. Experiments are performed with predictions up to one week into the future (Section 5.3.2). We experiment with different methods (Section 5.3.3) including effective, yet simple, Markov Models and models exploiting periodicity.

5.3.1 Venue categories

In this section we present an analysis focused on the venues where users check-in studying their temporal dynamics (Section 5.3.1.1) and the way users transition from one venue to the next (Section 5.3.1.2). The metadata associated with each Foursquare venue includes venue types. These are organized in a hierarchy with three levels. In our study we only use the most general layer of the hierarchy which contains nine venue types: 'Arts & Entertainment', 'Travel & Transport', 'Shop & Services', 'Food', 'Great Outdoors', 'Nightlife Spot', 'Residence', 'College & University' and 'Professional & Other Places'. The next layer of the hierarchy contains 259 venue types, which is too many for statistical and visualization purposes. Each venue can have multiple categories. In the rest of this section we use only the 'primary' category associated with every venue.

5.3.1.1 Temporal properties

Category frequency The percentage of check-ins for each category over our entire dataset is presented in Table 5.7. We observe that the categories with the highest number of check-ins are 'Food' and 'Shops & Services'. 'Travel & Transport' and 'Professional & Other Places' have high percentages as well, which mostly indicate that people use the service when traveling to different places and when they arrive at their office. 'Residence' has a significant percentage of check-ins as well, indicating that many users from our dataset check-in at their homes, probably when they leave or arrive. 'Nightlife Spot' and 'Arts & Entertainment' have low percentages as we would expect from a dataset that models regular day-to-day behaviour. When analysing LBSNs, there was a concern that many people use them only to show to their social network where they are going out. This appears not to be the case, which we ascribe to our filtering for frequent users who use the service more frequently and for a more general purpose.

Temporal frequency modelling We examine the temporal distribution of the check-ins for each category. We present the distribution of check-ins (denoted by Gold) on the entire month for the categories 'Professional & Other Places' in Figure 5.14 and for 'Nightlife Spot' in Figure 5.15.

We notice that the temporal frequency patterns are similar to those observed for periodic Twitter hashtags, e.g. in Figure 5.4. Consequently, in order to forecast future venue frequencies we use the modelling techniques introduced in Section 5.1.

Venue category	Percentage
Arts & Entertainment	3.7%
College & University	5.1%
Food	17.7%
Great Outdoors	6.8%
Nightlife Spot	3.7%
Residence	11.6%
Shop & Services	20.2%
Professional & Other Places	14.8%
Travel & Transport	15.9%

Table 5.7: Check-in frequency per venue category.

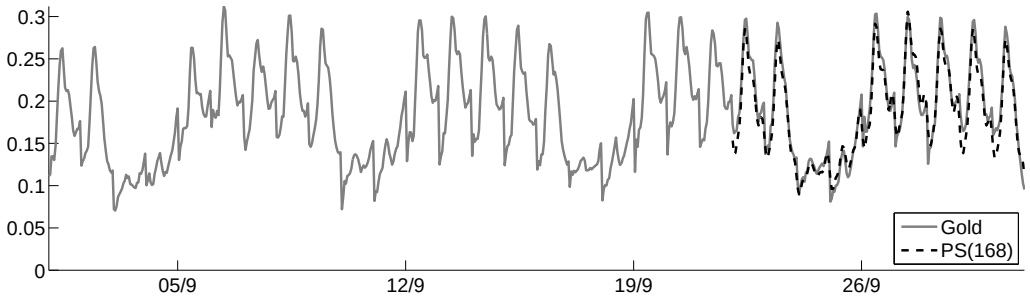


Figure 5.14: Check-in frequency for venue type 'Professional & Other Places' and forecast for the last 9 days.

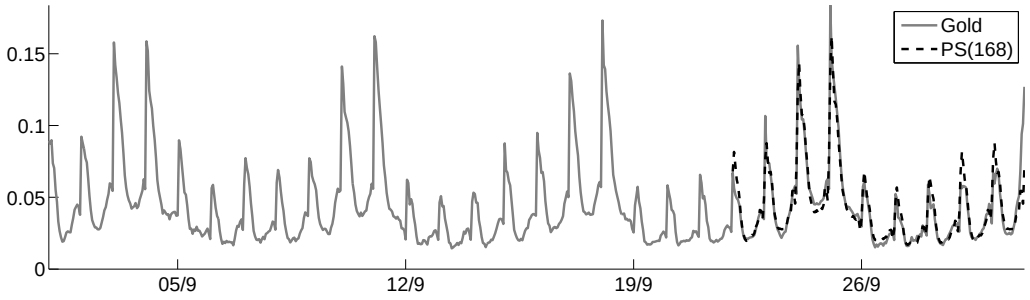


Figure 5.15: Check-in frequency for venue type 'Nightlife Spot' and forecast for the last 9 days.

We perform similar experiments on the 9 category time series using the same methods as in Section 5.2.4. We illustrate forecasts for two category frequency time series in Figures 5.14 and 5.15.

Both figures illustrate periodic patterns which are best fitted by the PS kernel with a weekly period. A daily drop in frequency indicates reduced activity during the nights. The category 'Professional & Other Places' consists mostly in office buildings. Activity here increases during weekdays with a low activity in weekends. During the days of the week, check-in patterns are consistent, with a spike during the morning,

when people arrive at work and a smaller one after lunch, if the users left for lunch and returned to work. The only weekday in our dataset with lower activity than the others is 5 September. The justification of this a U.S. holiday ('Labour Day'), where many of our users are based.

For the category 'Nightlife Spot', we observe a different frequency distribution with the same periodic patterns. While generally activity is low during the day and spikes in the evenings, we also notice a trend. Activity is lowest on Sundays and Mondays and starts growing until peaking on Friday and Saturday nights. This shows us that during weekdays, people tend to go out more in the evenings as the week progresses, reaching a high in the weekend nights.

Forecasting results are presented in Table 5.8. The venue category distributions were all best fit by periodic kernels (7 by PS and 2 by PER) with a period of 168 hours (one week). This explains the very high relative improvement over mean for the periodic methods. However, the **Lag+** model forecasts best. This is because the frequencies present significant consistency in periods and trends with little noise over the entire month. This is despite the number of check-ins being relatively low (at most 500 check-ins/hour for a category).

Lag+	**GP-Lin**	**GP-SE**	**GP-PER**	**GP-PS**	**GP+**
67.92%	0%	-104%	54.38%	61.61%	63.36%

Table 5.8: Average relative gain over mean (M) prediction for forecasting on the last 9 days month using the different models.

Temporal distribution While the previous experiments show and forecast the category check-in frequency at a certain hour, we are also interested in empirically studying the distribution of check-ins over venue categories. We aggregate check-ins over all weeks in our dataset as the weekly returning pattern is strongest (Figure 2.10). We look at activity on Saturdays as well as activity on all the weekdays combined as they showed similar temporal patterns for most categories. We present the check-in distributions in Figure 5.16 and Figure 5.17.

From the graphs we empirically observe interesting patterns. Check-ins into 'Professional & Other Places' are very frequent on weekdays and especially during the morning, when people arrive at work. This influx is preceded by an increase in check-ins to venues from the 'Travel & Transport' category. As we would expect, 'Colleges & University' follow the same trend as the 'Professional & Other Places' only at a smaller scale both on weekdays and Saturdays. The 'Residence' category has the

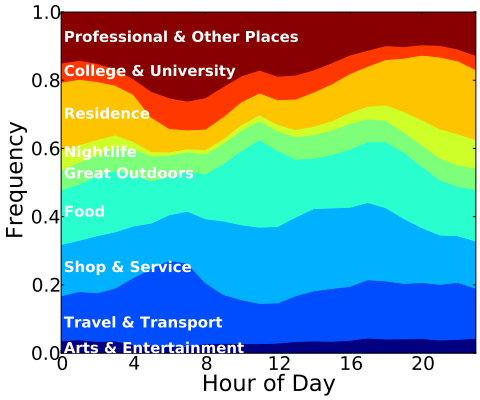


Figure 5.16: Weekday check-in distribution.

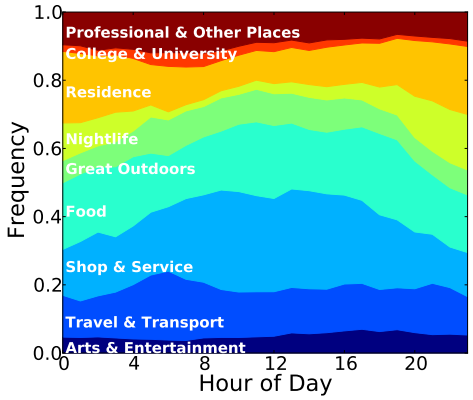


Figure 5.17: Saturday check-in distribution.

most check-ins in the morning (people wake up and check-in to their house) and in the afternoon (after 6pm) when people arrive home from work. For this category, the distribution is smoother on Saturdays but follows the same pattern. The 'Shops & Services' category has a consistent pattern as well, with increasing activity after lunch and decreasing as the night approaches. On Saturdays, 'Shop & Service' is the most important category during the day. As the night approaches, 'Nightlife Spot' venues become more popular, as during the day their frequency was the lowest.

We highlight that we only look at the time distribution of when check-ins occur which is different to the distribution of where the users from the dataset are. We do not assume that between two consecutive check-ins the user is located at his last registered location as we do not know that the users post their entire mobility data. The task of location prediction is in this case a partially observed inference problem. We expect that the venue type will influence how much time people actually spend there before moving to a new destination.

5.3.1.2 Transitions

We have focused so far on general, temporal and periodic analysis of venue categories. We now switch to a user-centered view by analysing individual mobility trails. We study transitions between venue categories. It was previously shown that Markov models of human behaviour perform well when predicting future locations [Song et al., 2003]. This means that there is an underlying structure in transitions. We present a heatmap of the transition probabilities from one venue category to another, aggregated over all the users in the dataset, in Figure 5.18.

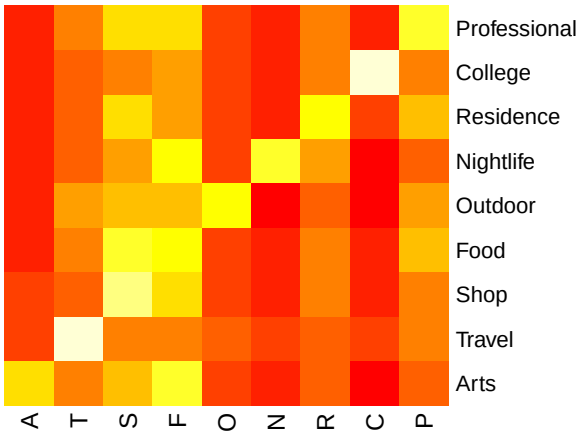


Figure 5.18: Transition diagram between categories. Lighter represents higher values. The transition source is on the vertical axis and the destination on the horizontal axis.

At first we notice the high frequency on the principal diagonal. This means, for most of the categories, that if a user is at a certain type of location, it is very likely for him/her to transition to a place belonging to the same category. This behaviour is most predominant for the 'Travel & Transport' and 'College & University' categories. This is expected, as people often visit many transport locations before reaching their destination and, if they are at a University building, it is very likely that they will visit another building from the campus next.

We also notice the lack of symmetry in the heatmap. This highlights that there are some transitions that are more likely to happen in one direction rather than the other. For example, it is very likely for people to go to eat (category 'Food') after an artistic event (category 'Arts & Entertainment'), while the opposite is less common. Other examples include the high probability for people to visit shops or to eat after work and in the lunch break or to go home after going out in the evenings. On the other hand, we also observe some reciprocal relations, such as the one between food and shopping activities, with all transitions with high values.

5.3.2 Experimental setup

We use the 'Frequent Users Dataset' introduced in Section 2.2.2. For the next location prediction task we use the following testing scenario. We train our models on the first three weeks (21 days) of data and test on the remaining days (9 days). After we obtain an average prediction percentage for each user, we then macro average the results and report this as the performance of the method.

5.3.3 Methods

Although [Eagle and Pentland, 2009] attempts to empirically show some temporal patterns, integrating them into prediction models has not been very successful [Lian and Xie, 2011]. This can be because the historical information was considered in conjunction with individual venues [Gao et al., 2012]. Very sparse representations are obtained and, as we collect more user data, the number of observed venues increases over time leading to a decrease in prediction performance. As also highlighted in [Lian and Xie, 2011], we must find a proxy for these venues that can be used for prediction. We use the category type information as this proxy, adding some semantics to user transitions. The advantage of using category information is that it is fixed to a limited number (i.e. 9), alleviating sparsity concerns, but still captures individual preferences and patterns.

We will use domain-independent methods for prediction. These methods take into account for prediction only the previous history of transitions of a user. The most common class of methods of this type is the Order-K Markov model, where we use as context the last K transitions of the user and find the most likely location where the user will transition next based on this context. This method allows us to learn simple transition rules for every user and predict the future location accordingly (e.g. after work a person will go to a food vendor, after a restaurant and a bar the person will likely go home or to another bar).

We want to integrate our knowledge about existing temporal patterns related to venue types. We have observed very strong patterns for each user and each category. The strongest was the weekly pattern, where a user transitions to the same venue category as the same day and time as one week before. Also strong was the daily pattern, where a user visits the same category type as the day before at the same time. We want to experiment with building these periodicities explicitly into a simple model in order to assess their predictive power.

Although other domain-independent methods exist, previous studies [Song et al., 2003] have shown that the improvement over the simple Markov based methods is marginal and, as our focus is on testing if incorporating the temporal returning patterns is beneficial, we chose not to implement them.

The methods we test are:

Most Frequent Category: We assign to each testing instance the category that was most frequent in the users' history. Note that this is equivalent to an Order-0 Markov predictor.

Markov-1: We assign to the testing instance the most frequent category based only on the last visited venue and its category. In case the transition was not observed in the users' history, we back-off to the 'Most Frequent Category'.

Markov-2: We assign to the testing instance the most frequent category based on the last 2 visited venues. In case the transitions were not observed in the users history, we back-off to the 'Markov-2' model.

The probability of the next check-in c_{n+1} at location l with an Order K Markov model is:

$$\begin{aligned} p_k(c_{n+1} = l|H) &= p(c_{n+1} = l|c_{n-k+1}, ..., c_n) \\ &= \frac{|\{c_r|c_r \in H, c_r = l, c_{r-j} = c_{n-j+1}\}|}{|\{c_r|c_r \in H, c_{r-j} = c_{n-j+1}\}|} \end{aligned} \tag{5.15}$$

where $H = \{c_1...c_n\}$ is the history of the previous check-ins.

Most Frequent Hour: We assign to the testing instance the category that was most visited by the user in the same hour of the day in its history. This model assigns the probability of the next check-in c_{n+1} at location l at time h as the probability of the location l occurring at time h in the previous check-in history:

$$p_{MFH}(c_{n+1} = l|H, t_{n+1} = h) = \frac{|\{c|c_r \in H, c_r = l, t_r = h\}|}{|\{c_r|c_r \in H, t_r = h\}|} \tag{5.16}$$

Most Frequent Day of Week and Hour: We assign to the testing instance the category that was most visited by the user in the same hour of the same day of the week in the training set. The probability of the model is similar to that of 'Most Frequent Hour', but in addition conditioning on the day of week.

5.3.4 Results

The results, measured as the percent of correct predictions, are shown in Table 5.9.

Method	**Accuracy**
Random Baseline	11.11%
Most Frequent Category (Markov-0)	35.21%
Markov-1 (with back-off to Markov-0)	36.13%
Markov-2 (with back-off to Markov-1)	34.21%
Most Frequent Hour	38.92%
Most Frequent Day of Week and Hour	40.65%

Table 5.9: Accuracy of different methods.

We observe that the results match our intuitions. We notice that the 'Most Frequent Category' baseline is quite high when compared to the other models. 'Markov-1'

with back-off performs slightly better which confirms that some information is gained by looking at the previous step of a transition. This improvement is however not that significant, indicating that user movement is governed also by other factors and patterns. 'Markov-2' performs slightly worst than the baseline. This confirms the previous results of [Song et al., 2003].

Incorporating explicitly the periodicity of user behaviour shows the best results. We observe that the 'Day of Week' pattern is strongest and obtains the best results, even if the history for a user is very restricted, basically to only 3 weeks of previously observed data. Actually, for almost half of the cases, we have used the backoff, which shows that the 'Day of Week' and 'Hour of Day' prediction is very effective when applicable. This method improves on all the Order-K Markov methods, showing that periodicity is a factor that has to be taken in account when studying human mobility patterns.

In all models, the overall macro-average is relatively low. The task of predicting future movements in the case of users which are not restricted to an age group, profession or geographical area is shown to be much harder then when using subjects that have the same characteristics such as lab students [Eagle and Pentland, 2009]. The results show the need for prediction algorithms that take into account temporal periodicities over different time intervals. However, some users have very different behaviours (e.g. holidays) in the testing period, switching between usage modalities. For example, over 8% of users have under 11% accuracy for the 'Most Frequent Category' method, meaning that these users have checked-in to their most frequent venue category from the training period less than random in the test set.

5.4 Conclusion

In this chapter we have presented a method that identifies complex temporal patterns in OSN data that go beyond temporal smoothness. Using Gaussian Processes, we have shown how to forecast future values based on past data, achieving state-of-the-art performance when compared to competitive methods. This modelling was effective not only for qualitative analysis, but also when incorporated downstream into applications, such as time-aware text classification. We have thus demonstrated that temporal context is important and worthwhile modelling in further applications.

While in this chapter we have focused more on behaviours that exist in large periods of time, many were shown to be unpredictable and ephemeral. These events

are likely caused as a response to external stimuli, most likely in the form of real-world events. In the next chapter we take a closer look at this effect and try to exploit sudden changes in word distributions in order to discover and describe their underlying topics and events.

Chapter 6

Detection of timely events

Social media text usage is often impacted by real-world events. This chapter aims to present methods that exploit these changes. Our goal is to detect events (i.e. newsworthy happenings) and track their evolution over time, both in volume and textual content. The underlying hypothesis is based on word co-occurrence: if two words appear in the same context more often than usual within a restricted time frame, then they are indicative of the same event. Clustering words within a time frame based on their co-occurrence would thus extract sets of weighted words that can uniquely define events from that respective time interval. In contrast, in large established corpora (e.g. Wikipedia) the contexts in which a word appears are stationary under the distributional hypothesis: words that occur in similar contexts tend to have similar meanings [Harris, 1954, Curran, 2004]). Clustering words based on their co-occurrence in static corpora would then uncover sets of semantically related words (i.e. topics) in the corpus. We also investigate co-occurrences in large time windows of social media data. In this case, we expect the temporal influence to be dampened.

We start with defining and analysing the temporal dynamics of word co-occurrences in OSN data in Section 6.1. We use the normalised version of the popular pointwise mutual information (PMI) metric for measuring co-occurrence between words in a dataset. A number of clustering methods are presented in Section 6.2 which use the co-occurrence score between words as a similarity metric in the clustering algorithm. We use the unsupervised spectral clustering algorithm. We experiment with different datasets in both time span and volume in Section 6.3. Focusing on clusters in restricted time frames, we show how to measure the magnitude and coherence of a cluster and, using a cluster centrality measure, automatically select relevant messages that can be presented as summaries for the clusters to end users. Clusters discovered at every time interval are hard to interpret longitudinally because of random

changes in cluster memberships and identifiability issues. In Section 6.4 we develop an evolutionary spectral clustering method which takes into account the linear temporal relationship between consecutive datasets with respect to time. The method treats time in a similar way to the forgetting factor in the online biconvex learning method presented in Section 4.4.2. This results in clusters which are consistent and identifiable over time. This way, we can study the dynamics of real world events as reflected by social media.

6.1 Word co-occurrence

Methods based on word co-occurrences have a long tradition in NLP and have been used successfully for applications ranging from sentiment analysis to thesaurus learning, collocation extraction and finding multiword expressions [Turney, 2002, Curran, 2004, Sag et al., 2002, Evert, 2005]. However in most cases, these scores have been computed over static corpora, ignoring any temporal variation. In this section we study the changing nature of word co-occurrences over time in a streaming setting using social media data. We hypothesise that the co-occurrence statistics of words will change over time based on real world events.

To illustrate the time evolution of word usage, consider the example of the word 'riot'. Using static corpora, e.g. Wikipedia, the highest co-occurring words would be syntactically or semantically related words (e.g. 'city', 'police', 'riots'). When computing the co-occurrence scores over a large dataset of two months of social media data, we obtain semantically related words (e.g. 'police', 'inciting', 'protesters') or words that are part of collocations (e.g. 'ra' from the band 'Ra Ra Riot', 'atari' from the group 'Atari Teenage Riot' or Atari's RIOT chip). However, when considering text from a restricted time interval from timely social media data, the set of most frequently co-occurring words changes over the course of 2011. Figure 6.1 displays the variation in frequency of the word 'riot' and the top co-occurring words in different time intervals. In January 'Egypt' and '#jan25' (the hashtag associated with the Egyptian revolution) commonly co-occur with 'riot' (due to the riots and the abundance of news and opinions about these events), while in February, 'riot' co-occurs more with words like 'Bahrain' where a different series of riots took place. Moreover, in small time frames, increases of co-occurrences of other seemingly unrelated words (e.g. 'Bieber') can be observed as a consequence of popular opinions or viral messages (e.g. 'If Bieber wins, we riot').

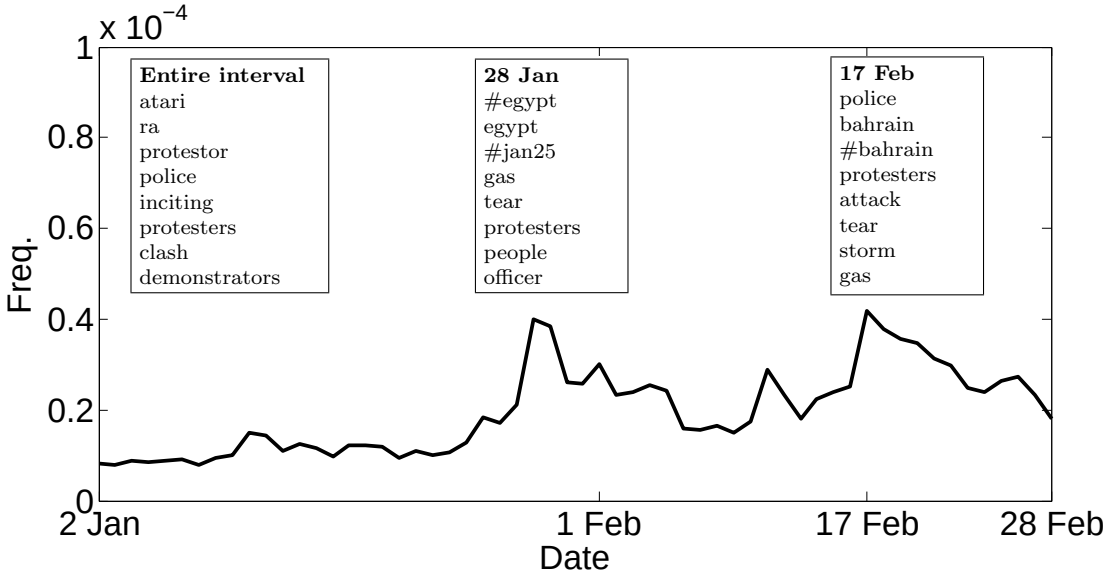


Figure 6.1: Temporal variation of the frequency of word 'riot' on Twitter and top co-occurring words in different time frames.

6.1.1 Pointwise mutual information

A standard method of uncovering associations between pairs of words is by computing the pointwise mutual information (PMI). The association score is usually obtained by computing the PMI value over a large but fixed corpora. In this section, we consider computing association scores over corpora of various time spans and sizes.

PMI is an information theoretic measure that indicates which words tend to often co-occur in a context. It measures how much the actual probability of a particular co-occurrence of words $p(x, y)$ differs from what is expected under the assumption of independence of the individual events, $p(x) \cdot p(y)$. A generalised formula for computing PMI is the following:

$$PMI(X, Y) = \alpha \cdot \ln \frac{p(x, y)}{p(x) \cdot p(y)} \tag{6.1}$$

where $p(x) = \text{freq}(x)/N$, $\text{freq}(x)$ is the count of word x and N is the total number of tokens in the dataset.

Note several issues when calculating PMI ($\alpha = 1$): the sensitivity to low count events and uninterpretable fixed values with a lacking upper bound. For example, if two words are perfectly correlated and $p(x) = p(y) = p(x, y)$ the PMI will have a value of $-\ln p(x, y)$. This makes the value dependent and inversely proportional to the number of times that word pair is observed. The value is hard to interpret numerically because it depends on the raw number of occurrences of the words.

A normalisation that addresses the interpretability issue was introduced by Church and Hanks [1990]. They present a version of the PMI, the Normalized PMI (NPMI) which is bounded in the $[-1, 1]$ interval and uses $\alpha = -\ln p(x, y)$. The interpretation of the values of the NPMI is natural and fixed for any word pair. A positive NPMI value denote word pairs that co-occur more often than predicted by change, and negative values less often. Bouma [2009] shows that even though NPMI differs from PMI, the information they capture is essentially very similar. However, PMI and, to a smaller extent, NPMI are sensitive to rarely occurring events [Manning and Schütze, 1999] because of the uncertainty of the estimate for the true value of $p(x, y)$. It is thus common to introduce a threshold in the minimum number of co-occurrences for which the NPMI value is computed. In our examples, we chose this to be 5. We will use NPMI as the word co-occurrence measure and consider this as a similarity measure between words computed over a specific dataset for our subsequent experiments.

In a streaming setting, the main computational issue for computing the PMI is the numerator in Equation 6.1. If the vocabulary size is large, which is usually the case, updating these values becomes computationally infeasible. Van Durme and Lall [2009] have shown the impracticality of perfect online PMI computation and presented methods to compute these values with high expected accuracy. In order to avoid these issues, we chose to segment the corpus into fixed-length time intervals (e.g. one hour, one day), discretising time. If this interval is too short, very few co-occurrence values are significant. Conversely, if the time interval is large, then short lived events might not be identified. NPMI computation is easily parallelisable, which we exploit by using a cluster running Apache Hadoop.

The association measure of words is normally indicative of collocations. Given data from a narrow time frame, however, we expect that words co-occur often if they are indicative of the same event that was mentioned in multiple texts. By clustering words based on co-occurrences we expect to find the terms that can reliably and uniquely characterise an event. Table 6.2 presents a list of word pairs with very high NPMI scores as computed over a large two months social media dataset. These consist mainly of frequent collocations (e.g. names of singers, TV series) where each word rarely occurs without the other one. In Table 6.1 we present a selected list of word pairs with very high NPMI computed over an hour long social media dataset which show high co-occurrence values for terms in current news at that time.

Word1	Word2	NPMI	Description
baghdad	bombs	0.705	Baghdad bombings
troops	ufc	0.704	UFC Fight for the Troops show
cameras	spotted	0.668	LG G-Slate tablet camera spotted
iran	nuclear	0.646	Iran nuclear ambitions
djokovic	quarters	0.641	Djokovic in Australian Open quarterfinals at tennis
fighter	stealth	0.639	China unveils stealth fighter
motorola	launching	0.636	Motorola XOOM announced for launch on 17th February
jets	steelers	0.632	New York Jets - Pittsburgh Steelers Football match
activist	arrested	0.614	Yemen arrests anti-government activist
protests	tunisia	0.605	Protests in Tunisia

Table 6.1: Selected top NPMI values for 23 Jan, 9-10am. Word1, Word2 are in alphabetical order.

Word1	Word2	NPMI	Description
feng	shui	0.734	philosophical system
#sunrise	#sunset	0.727	weather hashtags
phineas	ferb	0.723	TV series
avenged	sevenfold	0.723	music band
erykah	badu	0.722	singer
mardi	gras	0.722	event
tinie	tempah	0.721	singer
gwyneth	paltrow	0.721	actress
battlestar	galactica	0.721	TV series
estee	lauder	0.720	company

Table 6.2: Top NPMI values over the entire two months of the Twitter Gardenhose dataset. Word1, Word2 are in selected order for enhanced interpretability.

6.1.2 Temporal evolution

Our assumption is that the co-occurrence distribution of words changes in time as a consequence of real-world events. We first analyse pairs of words relating to events (event description is in Section 6.3.3). For four such word pairs we display the temporal evolution of co-occurrence counts in Figure 6.2 and of the NPMI values in Figure 6.3. From Figure 6.2 we observe that pairs like 'Taco Bell' and 'Toyota recall' co-occur often in a daily pattern. For 'Taco Bell' this happens mostly because U.S. users comment on the company's products every day. The other four pairs mostly co-occur around events related to them, with a decay as the relevance of the event diminishes. In the 'Taco Bell' case, this trend is combined with the daily pattern. In contrast, when analysing NPMI from Figure 6.3, all the pairs have high values at the start of the related event keeping a non-zero value until the event looses its importance almost entirely. This shows that NPMI values remain high even when

the co-occurrence counts are relatively low. This would allow a clustering algorithm applied to co-occurrences to uncover even events with lower popularity. As the NPMI can also have negative values and can be close to zero due to noisy estimates, it is prudent to set a threshold below which NPMI values are set to zero [Turney and Pantel, 2010].

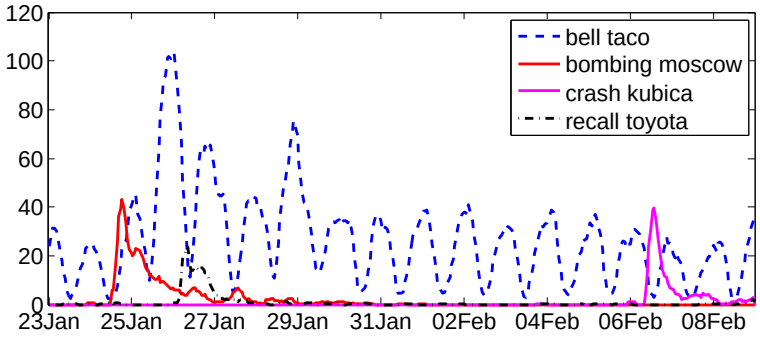


Figure 6.2: Temporal variation of co-occurrence pairs.

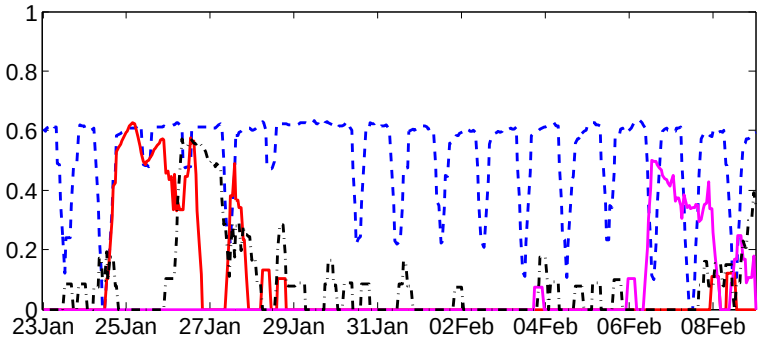


Figure 6.3: Temporal variation of co-occurrence values (NPMI).

6.1.3 Applications

The study of word co-occurrences has a long tradition in Natural Language Processing. Measures of co-occurrence have been first studied by Fano [1961] and Dunning [1993]. They have been used for finding collocations or multiword expressions in documents [Sag et al., 2002, Evert, 2005], for weighting vectors for measuring distributional semantic similarity [Turney and Pantel, 2010] or for finding the sentiment polarity of words [Turney, 2002]. More related to our experiments, Newman et al. [2010] obtain by using PMI the best performance for measuring topic coherence.

6.2 Methods

In this section we introduce the methods used to identify events and topics in social media data. These methods are based on grouping words into clusters which are used to characterise events or topics. We will refer to a 'topic' as having a broader meaning [Lin et al., 2011], e.g. 'football' or 'U.K. politics', combining semantically related concepts. We define an 'event' with a more restricted meaning, similarly as the term 'topic' in the context of Topic Detection and Tracking [Allan, 2002]: 'a set of news stories that are strongly related by some seminal event that triggers a topic', e.g. a single football match or a political debate. There is however a fine line between these concepts as some events can be very broad and thus similar to a topic.

Generally, two different classes of clustering methods exist: hard or soft clustering. In hard clustering, the items are divided into mutually exclusive clusters. Conversely, in soft clustering an item can belong to multiple clusters and has a probability distribution across all clusters. This indicates the association strength between the item and each cluster.

The latter set of methods contain topic models (see Section 4.2.4) which identify distributions over vocabularies based on document level co-occurrences called 'topics', effectively performing a soft clustering on the vocabulary. After initial experimentation using online LDA we found uninterpretable results. This is due to the short message length and frequent foreign terms co-occurring within a tweet.

In this chapter, we develop and experiment with two hard clustering methods: K-means clustering [MacQueen, 1967] and spectral clustering [Ng et al., 2002]. These make use of the co-occurrence scores between words as a similarity measure in order to create clusters that reflect events/topics. Some assumptions can be made about the data and the behaviour we aim to model. Our goal is to preserve clusters that contain pairs of highly co-occurring words and their local neighbours. Due to the distributional similarity hypothesis, not all words need to have high similarities with every other word in order to refer to the same event/topic as they might occur interchangeably. We thus aim for clusters to include words that may not co-occur, but are distributionally similar (e.g. 'recall' and 'recalls'; 'Moscow' and 'Domodedovo' - i.e. Moscow's airport). The size of the clusters is unknown and we expect it to depend on the event type. The number of clusters can also vary dramatically based on the time of day and the day itself (e.g. weekday vs. weekend).

6.2.1 K-means clustering

Hard clustering algorithms aim to partition n items (i.e. words in our case) into k groups or clusters. Items in each cluster should be similar whilst being different to items in all the different clusters. This can be formulated as a combinatorial optimisation problem by defining a loss function:

$$W(\mathcal{C}) = \sum_{i=1}^{k} \sum_{w,w' \in \mathcal{C}} d(w, w') \tag{6.2}$$

where $\mathcal{C} = \{C_i\}_{i=1}^{k}$ is the set of k clusters and d is a disimiliarity function.

This is called between-cluster point scatter and minimising $W(C)$ for all cluster assignments is not computationally feasible even for low values of n and k. For identifying the optimal clusters, an approximate algorithm needs to be developed. Most common are algorithms based on iterative greedy descent such as the K-means algorithm [MacQueen, 1967]. These algorithms change cluster assignments at each iteration such that the criterion is always lowered until convergence or a limit in the number of iterations is reached. However, reaching a global optimum is not guaranteed, but only a local one.

Algorithm 2 Partitioning Around Medoids

$k \in \mathbb{R}$ ▷ Number of clusters (fixed)
$\mathcal{D} = \{w_1, ..., w_n\}$ ▷ Data items
$D \in \mathbb{R}^{n \times n}$ ▷ Pairwise disimiliarity matrix
procedure PAM($k, \mathcal{D}, D$)
    randomly initialise $M = \{m_1, ..., m_k\}, M^* = \varnothing$ ▷ Randomly initialise medoids
    while $M! = M^*$ **do**
        $M^* \leftarrow M$
        for $w_i \in \mathcal{D}$ **do**
            $C_{\text{argmin}_k D(i,m_k)} \leftarrow w_i$ ▷ Assign each data point to the cluster of its closest medoid
        end for
        for $m_i \in M$ **do**
            for $w_j \in C_i$ **do**
                if $W(M \setminus \{m_i\} + \{j\}, \mathcal{C}) < W(M, \mathcal{C})$ **then**
                $m_i^* \leftarrow j$ ▷ Check in each cluster i if a 'better' medoid exists
                end if
            end for
        end for
    end while
end procedure

The K-means algorithm is a centroid-based clustering algorithm that aims to partition the n items such that each item belongs to the cluster with the nearest mean, serving as a prototype of the cluster. However, in computing the distance between items, K-means algorithm uses the Euclidean distance between the items, which is not suitable for any task. The K-medoids algorithms is an alternative to the K-means algorithm where the cluster prototype is restricted to be one of the items, i.e. the medoid instead of the mean. In this way, a custom distance function can be used in clustering, given as a distance (or similarity) matrix between each pair of data items. Because of this, K-medoids is considered to be more robust to noise and outliers in comparison to K-means. The most common solution to K-medoids clustering is the Partitioning Around Medoids (PAM) (Algorithm 2) [Theodoridis and Koutroumbas, 2008]. The function to optimise for this algorithm takes the form:

$$W(M, \mathcal{C}) = \sum_{i=1}^{k} \sum_{w \in C_i} d(w, w_{m_i}) \tag{6.3}$$

where $M = \{m_1, ...m_k\}$ are the k medoid indices.

6.2.2 Spectral clustering

Spectral clustering [Shi and Malik, 2000, Ng et al., 2002] is a clustering algorithm that has been shown to achieve state-of-the-art performance for a range of tasks, from image segmentation [Shi and Malik, 2000] to community detection [Smyth and White, 2005]. Besides its performance, it is also appealing because of its theoretical grounding in spectral graph theory [Chung, 1997]. This algorithm treats the clustering problem as one of graph partitioning on the similarity graph between data points. Following [Ng et al., 2002], the objective of (normalised) spectral clustering is to minimise the normalised cut (*Ncut*):

$$W_{Ncut}(\mathcal{C}) = \frac{1}{2} \sum_{c=1}^{k} \frac{\sum_{i \in C_c, j \notin C_c} s_{ij}}{\sum_{i,j \in C_c} s_{ij}} \tag{6.4}$$

where $S = \{s_{ij}\}_{i,j=1}^{n}$ is the adjacency matrix of the similarity graph. The *Ncut* objective is to find the best graph partition such that similarities between partitions are low while the similarities within each partition have high values. The normalisation of each cluster with the sum of the within cluster similarities is also performed in order to avoid degenerate solutions with clusters of size one. This way, we aim to have clusters with similar volumes. Another view of the *Ncut* objective is to identify

a partition such that a random walk weighted by s_{ij} in the graph seldom changes cluster memberships. However, solving the *Ncut* problem is NP-hard.

If we define:

$$D = \text{diag}\left(\{\sum_{j=1}^{k} s_{ij}\}_{i=1}^{n}\right) \tag{6.5}$$

$$L = D - S \tag{6.6}$$

$$H = \{h_{ij}\}_{i,j=1}^{n} = \begin{cases} 1/\sqrt{\sum_{k_1,k_2 \in C_j} s_{k_1 k_2}} & \text{if } w_i \in C_j \\ 0 & \text{otherwise.} \end{cases} \tag{6.7}$$

the *Ncut* problem can be reformulated as:

$$\min_{C} \text{Tr}(H'LH), \text{ subject to } H'DH = \mathbb{I}. \tag{6.8}$$

By relaxing the constraint on the discrete structure of H from Equation 6.7 and by defining $T = HD^{1/2}$, the objective becomes:

$$\min_{C} \text{Tr}(T'D^{-1/2}LD^{-1/2}T), \text{ subject to } T'T = \mathbb{I}. \tag{6.9}$$

where $L_{sym} = D^{-1/2}LD^{-1/2}$ is called the normalised graph Laplacian.

This is a trace minimisation problem with the solution T having as columns the first k eigenvectors of L_{sym}, corresponding to the smallest eigenvalues and ignoring zero values. A final discretisation step is then performed on the matrix T in order to transform it to the original structure of H and obtain a hard clustering of the data.

Intuitively, the spectral clustering algorithm projects the data points via Singular Value Decomposition (SVD) into a reduced space which aims for maximal separation of clusters. This way, spectral clustering is useful when data dimensionality is high. It is also particularly useful when the clusters are hard to be discovered using a spherical metric, as is the case, for example, with K-means.

Normalised spectral clustering [Ng et al., 2002] is presented in Algorithm 3. We start with the similarity matrix S and with a choice of the number of clusters k. After computing the diagonal matrix D and the graph Laplacian L, SVD is applied to identify the first k eigenvectors of the Laplacian. On the reduced matrix T, K-means clustering is performed in order to identify the cluster memberships.

A good clustering relies on a good similarity matrix which best reflects the connections between objects. Sparsity of this matrix is another desirable property as the eigensolver can produce faster results with sparser matrices. Most of the work in applying spectral clustering is performed to build the optimal similarity graph

Algorithm 3 Normalised Spectral clustering

$k \in \mathbb{R}$ ▷ Number of clusters (fixed)
$S \in \mathbb{R}^{n \times n}$ ▷ Pairwise similarity matrix
procedure NSPECTRAL(k, S)
    $L_{sym} \leftarrow \mathbb{I} - D^{-1/2} S D^{-1/2}$ ▷ Compute graph Laplacian
    $U = \{\boldsymbol{u}_i\}_{i=1}^{k} \leftarrow SVD(L_{sym}, k)$ ▷ Get first k eigenvectors
    $T = \{t_{ij}\}_{i,j=1}^{k}$, $t_{ij} \leftarrow u_{ij} / \left(\sum_k u_{ik}^2\right)^{1/2}$
    $\mathcal{C} \leftarrow KMeans(\boldsymbol{t}_1, ..., \boldsymbol{t}_n)$ ▷ Run K-means on the reduced space
end procedure

which best reflects the relationships in the data. The application of spectral clustering methods in NLP has been limited because of increased storage space and runtime when faced with large-scale text datasets [Lin and Cohen, 2010]. Further details, proofs and a discussion on the approximation to the *Ncut* solution are presented in [Von Luxburg, 2007].

6.2.3 Evolutionary spectral clustering

When analysing temporal data, the clusters at each time interval are expected to be similar to the ones from the previous time interval and smoothly vary over time. Our assumption is that the data is computed over a restricted time window such that it can be considered stationary. If performing independent clusterings for each time interval there is no explicit relationship between clusters. This causes problems as the clustering results are sensitive to noise and the results can be unstable and inconsistent with the ones from the previous time step. Another issue is the identifiability of clusters through consecutive time steps.

Evolutionary clustering algorithms [Chakrabarti et al., 2006] have been developed for clustering objects that evolve over time and can be used to track changes over time. This aims to reflect long-term trends while being robust to short-term variations by adding a temporal smoothness penalty to a static clustering method. Further methods for evolutionary spectral clustering were developed by Xu et al. [2010], which build on the work of Chi et al. [2007] and which we present below.

The intuition of the algorithm is that a current partition $\mathcal{C}^t$ at time t needs to both partition the current dataset well and also to perform well in partitioning the data at the previous time step $t-1$. This way, we seek partitions that are also consistent with past data. The alternative to this is to impose $\mathcal{C}^t$ to be similar to $\mathcal{C}^{t-1}$ in addition to performing a good partitioning of the data. The objective of the

evolutionary spectral clustering at time t is thus a convex combination of the cost at times t and $t-1$:

$$\begin{aligned} W_{EvNcut}(\mathcal{C}) &= \alpha \cdot W_{Ncut}(\mathcal{C}^t) + (1-\alpha) \cdot W_{Ncut}(\mathcal{C}^{t-1}) \\ &= \alpha \cdot \frac{1}{2} \sum_{c=1}^{k} \frac{\sum_{i \in C_c, j \notin C_c} s_{ij}^t}{\sum_{i,j \in C_c} s_{ij}^t} + (1-\alpha) \cdot \frac{1}{2} \sum_{c=1}^{k} \frac{\sum_{i \in C_c, j \notin C_c} s_{ij}^{t-1}}{\sum_{i,j \in C_c} s_{ij}^{t-1}} \end{aligned} \tag{6.10}$$

where $0 \leq \alpha \leq 1$ is a forgetting factor which controls the amount of smoothing to be applied.

By relaxing the objective in a similar way as in the previous section, we obtain:

$$\min_C \mathrm{Tr}(T'(\alpha D_t^{-1/2} L^t D_t^{-1/2} + (1-\alpha) D_{t-1}^{-1/2} L^{t-1} D_{t-1}^{-1/2})T), \text{ subject to } T'T = \mathbb{I} \tag{6.11}$$

We have the same trace optimisation problem as in Equation 6.9 for which the solution is the matrix of the first k eigenvalues of $\alpha D_t^{-1/2} L^t D_t^{-1/2} + (1-\alpha) D_{t-1}^{-1/2} L^{t-1} D_{t-1}^{-1/2}$. Considering $L = D - W$, we can simply define the similarity matrix at time t for evolutionary clustering as being:

$$S_*^t = \alpha S^t + (1-\alpha) S^{t-1} \tag{6.12}$$

The problem of choosing the α adaptively at each time step is approached by Xu et al. [2010]. Because it is smoothed, the new similarity matrix S_*^t has lower variance than S_t. However, this might induce a bias leading to a bias-variance trade-off problem. An approach similar to [Ledoit and Wolf, 2003] for shrinkage estimation of covariance matrices is used for choosing α. Its objective is to optimise the square Frobenius norm of the difference between the true underlying similarity matrix and the S_*^t. As the true similarity matrix is unknown, all unknowns are replaced with sample equivalents [Schäfer and Strimmer, 2005]. Practically, this consists in iterating until convergence between obtaining an approximate α^* based on the cluster assignments and updating the assignments while fixing α^*. Full details of the algorithm can be found in [Xu et al., 2010]. We note that this is the first time evolutionary clustering algorithms have been applied in an NLP task.

6.3 Experiments

In this section we present the datasets (Section 6.3.1) and the experiments (Sections 6.3.3, 6.3.4) we conduct in order to evaluate our clustering algorithms. We also introduce the cluster measures used in our experiments (Section 6.3.2). Two different types of evaluation for clustering exist: internal and external [Manning et al.,

2008]. Internal evaluation aims to evaluate and compare an internal criterion of the clustering algorithm, e.g. the objective function. However, good performance on the internal evaluation might not correspond in better experimental results if the assumptions (e.g. similarity metric) do not match the desired goal of the clustering. External evaluation aims to evaluate how well the results of the clustering algorithm match its desired application. While this form of evaluation is more direct, it is also very hard to perform at a large scale as human evaluation is costly. An alternative for human evaluation is to provide a set of gold standard classes e.g. sets of words that belong in the same cluster. An external criterion can evaluate how well the clustering results map to the gold standard classes. We perform automatic internal and external evaluation of our methods in Section 6.3.3.1 in order to compare algorithms and their setups. The rest of the experiments are analysed qualitatively and evaluated using human judgements.

6.3.1 Data

We use multiple datasets in both volume, content and time span for our experiments (see Section 2.1.3). First, we use the Twitter Gardenhose dataset from 1 January – 28 February 2011. Because of the large volume of data (∼9 million messages daily), computing the co-occurrences and clusters is performed using one hour time intervals. The second corpus is the Austrian Politics spanning one year: from September 2012 to September 2013. This has the advantage of being focused on specific real-world events (i.e. relating to Austrian politics) and thus containing fewer conversation and personal messages which often represents noise for the goal of event detection. Because the data is of a smaller volume (∼1.8 million messages in total), for the temporal experiments each partition will consist of tweets authored across one day intervals. We also experiment with computing co-occurrences and clusters over each entire corpus in order to validate our hypothesis.

For text processing, we have tokenised and deduplicated all the tweets using the methods presented in Chapter 3. Removing duplicate tweets is important as these messages bias the co-occurrence values, artificially inflating the counts for the terms therein. For computing the NPMI, we consider that two words co-occur if they are part of the same message. Commonly, co-occurrences were computed over a word window. We chose our method because tweets are short (∼6 vocabulary tokens) and we can assume that they refer only to one topic. The vocabulary consists of the most frequent 50,000 tokens. In the second dataset, the vocabulary is formed of words with at least 10 occurrences in the dictionary (48,946 in total).

6.3.2 Cluster measures

In order to measure the 'quality' of a cluster we compute for each word w an average coherence score with respect to the words in its assigned cluster C using the original similarity scores (even if we will use other variations of it in clustering):

$$Q_w = \sum_{x \in C} \frac{\text{NPMI}(w, x)}{\binom{|c|}{2}} \tag{6.13}$$

The coherence of cluster c is defined as the average coherence of its words:

$$Q_C = \frac{\sum_{w \in C} Q_w}{|C|} \tag{6.14}$$

We define the magnitude of a cluster as the average co-occurrence frequency across all word pairs in the cluster. In order to measure which tokens are more representative for a cluster we compute for each word a centrality measure using the following formula:

$$\text{Ce}_w = \frac{\sum_{x \in C} \text{NPMI}(w, x)}{|C| - 1} \tag{6.15}$$

This scores highly words with high co-occurrence scores with all the other tokens which are expected to be more central concepts in its cluster.

6.3.3 Gardenhose experiments

Our first set of experiments are conducted on the Gardenhose dataset. This consists of a very large set of unfiltered tweets which contains, other than timely event related messages also many conversational or general tweets. We aim to see if clustering in small time windows (here one hour) can uncover events and if it can filter the noise arising of conversational messages. We first present in Section 6.3.3.1 a quantitative analysis which aims to compare the different clustering methods, show how they improve over baselines and present how the parameters of the algorithms can be adjusted automatically. Qualitative experiments in Section 6.3.3.2 present and evaluate events discovered using word co-occurrence statistics and clustering. We also perform aggregate experiments over the entire two months of data. Given the general purpose and large volume of the dataset, we expect word co-occurrences not to be affected by short term perturbations and have high co-occurrence scores with semantically related words.

6.3.3.1 Quantitative analysis

First, we quantitatively compare K-means (denoted **K**) and spectral (denoted **S**) clustering and analyse the sensitivity to different parameter settings. We perform 24 individual evaluations on each hour of a randomly selected day of the dataset (i.e. 23 January 2011) and report the average scores across all hours. The most important parameter is the number of clusters (k). Beyond this, also important is the underlying similarity graph, which should be sparse for efficiency reasons. We experiment with a few setups, each building on the previous. First, we discard all the NPMI values less than 0.3. From the resulting graph we discard all items that do not belong to the largest connected component, as these components typically represent trivial clusters. This way, we remove both common words that are unspecific for any event/topic and words which are correlated only to a few others. We denote this setup with the **-f** suffix e.g. K-f is K-means with this setup. For comparison purposes, this reduced vocabulary is used in all experiments for the respective hour. From the resulting graph, we build a mutual nearest-neighbourhood graph with $n = 50$[1] (denoted **-r**). We experiment with 'spreading' the values in the $[0, 1]$ interval by applying a Gaussian similarity function[2] with $\sigma = 0.3$ (denoted **-s**). Emphasising higher NPMI scores was shown to produce better results in some applications [Aletras and Stevenson, 2013]. K-means with no pruning of the NPMI values is denoted as **K** and a random partition as **R**.

As a means of internal cluster evaluation we use the total coherence score for a clustering. This is defined as the average across all word coherence scores (see Equation 6.13) and corresponds to the objective of K-means clustering from Equation 6.2. The results are presented in Table 6.3. We note that scores are not comparable across different number of clusters k, as larger clusters tend to have lower scores. We discover first that K-means has better results for lower number of clusters, but this changes in favour of spectral clustering when increasing the number of clusters above 300. This is somewhat expected: the K-means algorithm collapses many words into large clusters ($k = 500, \sigma = 131$) because they have similar values (close to 0) for the majority of dimensions. Spectral clustering avoids this problem ($k = 500, \sigma = 16$) by performing clustering on a reduced space that provides a better separation. Under this measure, the three setups have similar results, with **(s)** having the advantage of a lower computational cost.

[1] Lowest value that keeps the graph connected as recommended by Von Luxburg [2007]

[2] $gs(x) = 1 - exp(\frac{-x}{2 \cdot \sigma^2})$

For automatic external evaluation, a gold standard set of word clusters is needed. For this, we consider that the word, its equivalent hashtag and its plural should appear in the same cluster (e.g. 'packer', '#packer', '#packers'). These pairs also are likely to have a very small NPMI value, because they rarely co-occur in the same tweet (Twitter users use the hashtag to substitute the word). In our vocabulary there are in total 2009 pairs and 114 triples (due to pruning by largest connected component, not all are present in every clustering). We consider purity [Manning et al., 2008] as measure:

$$\text{purity}(\mathcal{C}, \Omega) = \frac{1}{n} \sum_{k} \max_{j} |C_k \cap \omega_j| \tag{6.16}$$

where Ω is the set of gold standard clusters. The score is also presented relative to a random baseline that keeps the cluster sizes fixed and randomises the assignments as in [Bamman et al., 2013], as the size distribution influences the purity score. For example, a partitioning which has a very large cluster containing all gold standard pair achieves maximum purity score but is not interpretable. The scores are presented in Table 6.4, showing that all our models obtain better performance even when confronted with this difficult task, with spectral clustering (**S-s**) with $k = 500$ having the best improvement compared to the baseline. All results compared to the random baseline are statistically significantly better, $p < .01$. Because spectral clustering showed consistently better results, we only use this algorithm in the qualitative experiments sections. We highlight that the experiments which lead to the optimal parameter settings for experiments to different factors such as vocabulary size or data volume. As a consequence, similar quantitative experiments need to be conducted in order to obtain tailored parameter settings.

k	**R**	**K**	**K-f**	**K-r**	**K-s**	**S-f**	**S-r**	**S-s**
50	0.005	**0.015**	0.013	0.012	0.012	0.010	0.010	0.010
100	0.005	**0.023**	0.020	0.020	0.020	0.016	0.016	0.016
200	0.005	**0.035**	0.034	0.033	0.033	0.030	0.030	0.030
500	0.004	0.066	0.064	0.064	0.064	0.087	0.087	**0.088**

Table 6.3: Average word coherence. Bold numbers show best performance. Results not comparable across different values of k.

6.3.3.2 Qualitative analysis

In this section, we present a qualitative analysis of our method. We use the single model that yielded best results in the previous Section, i.e. spectral clustering with 500 clusters on the reduced matrix and with similarity function (**S-s**).

k	K	K-f	K-r	K-s	S-f	S-r	S-s
50	0.20 ↑ 11%	0.13↑ 11%	0.11↑ 11%	0.11↑ 9%	0.10↑ 9%	0.12↑ 9%	0.12↑ 8%
100	0.32 ↑ 9%	0.23↑ 10%	0.20↑ 9%	0.19↑ 7%	0.23↑ 11%	0.25↑ 11%	0.25↑ 11%
200	0.48 ↑ 8%	0.37↑ 9%	0.33↑ 8%	0.32↑ 8%	0.42↑ 15%	0.43↑ 14%	0.43↑ 14%
500	0.68 ↑ 6%	0.59↑ 7%	0.55↑ 7%	0.56↑ 7%	0.64↑ 16%	0.65↑ 16%	0.65↑ 16%

Table 6.4: Purity on labeled word pairs. Percentage shows improvement over a controlled random baseline.

We expect many clusters to be non-indicative of any event as many words are only conversational or are otherwise not very important to any specific event. Such words would appear in very different contexts and consequently will have low co-occurrence scores. We only keep clusters with a high coherence score (see Equation 6.14), empirically set for the current experiments to a value of 0.15. This way, we also control for the different number of events in a dataset. On average, the number of clusters within one hour with a coherence score above the 0.15 threshold is 175.6 with a standard deviation of 41. After reducing the full matrix to the one used in clustering, we have on average 7858 words as input to the clustering algorithm.

Cluster quality Our first experiment aims to provide an overview of the contents of the clusters. We randomly chose a date and hour from our dataset (24 January 2011, 9-10pm G.M.T.) and asked two independent annotators to judge if and how relevant each cluster is to an event on that hour. The Inter-Annotator agreement (IAA) is 0.67 and Cohen's kappa $\kappa = 0.53$ (significant at $p < .01$). Results are presented in Table 6.5 showing that most of the clusters are indeed related to events.

Event related	Partially related	Not related	Spam
44.27%	20.99%	18.32%	16.41%

Table 6.5: Cluster quality judgements over 131 clusters averaged over two independent annotators.

Sample clusters We start by presenting a random set of clusters of different number of words from this hour of the dataset in Figure 6.6. The top clusters in terms of coherence are presented in Table 6.7 and in terms of magnitude are displayed in Table 6.8. We will refer to clusters in text with their topmost word from the table.

We observe that, for the majority of clusters, we can deduce the key information about the event that it describes. For example, cluster **'#sotu'** refers to discussion about the 'State of the Union' address[3] (happening on the next day, 25 January 2011)

[3] `http://www.whitehouse.gov/state-of-the-union-2011`

show	report	action	stay
name	gig	grab	says
tv	cnn	joint	reuters
ideas	ceo	military	continue
eric	producer	discuss	bob
awarded	google's	rebel	chief
schmidt		countries	tuned
outgoing		region	executive
rumored		rumors	remain
		defence	aig
0.31	**0.35**	**0.31**	**0.24**

6	shower	ap	making
review	broke	points	lord
driving	daughter	tech	taylor
series	grandma	jones	judge
gear	jimmy	average	trial
active	abc	solid	defense
previous	someone's	closing	claims
published	reveals	shy	false
o'clock	rivers	stocks	jury
bmw	melissa	closes	enses
	entering	dow	investigating
		12,000	
0.22	**0.25**	**0.35**	**0.29**

#sotu	goal	football	f
skills	score	sports	fair
held	chelsea	sky	ca
protect	goals	female	county
americans	blues	andy	pressure
millions	scored	comments	field
illegal	scores	keys	falling
secure	41	gray	mph
workers	56	richard	calm
laws	#football	assistant	winds
lottery	2-0	sack	mb
border	bolton	rant	wa
@barackobama	#cfc	sacked	kt
visa	4-0	sian	nc
existing	#chelsea	sexist	wi
#s	#epl	aimed	mn
educational	drogba	ironing	reporting
immigration	reebok	sexism	clouds
overly	davies	apologise	automatic
mandatory	anelka	referee	regional
employers	@chelseafc	presenter	overcst
	cech	massey	southwest
	0-4		hu
	malouda		northwest
	ramires		terminal
	#bwfc		72
0.22	**0.26**	**0.29**	**0.23**

Table 6.6: Example of clusters and coherence scores for 24 January 2011 9-10pm G.M.T.

behind	weather	cheaper	works	sea
john	chill	replacement	manager	pure
tweetdeck	across	rubber	bears	boat
medium	canada	originally	coach	pushed
keen	recent	doh	general	breeze
techcrunch	brave		smith	depth
mogul	ch		jerry	probe
tctv	canadians		extension	flowing
	coldest		angelo	curiosity
	warnings		lovie	
0.53	**0.42**	**0.42**	**0.38**	**0.38**

Table 6.7: Most coherent clusters and coherence score for 24 January 2011 9-10pm G.M.T.

while **'goal'** refers to details about the Bolton Wanderers – Chelsea English Premier League football match.[4] We notice that both clusters contain key related terms, in the case of the **'#sotu'** cluster topics that will probably be addressed such as 'immigration', 'employers, 'laws', 'protect and in the case of the **'goal'** cluster the names of the football teams ('bolton', 'chelsea'), all the four goalscorers ('drogba', 'malouda', 'anelka', 'ramires') as well as both teams' hashtags ('#cfc', 'bwfc'). Conversely, the

[4] `http://www.telegraph.co.uk/sport/football/competitions/premier-league/7943954/Bolton-Wanderers-v-Chelsea-live.html`

down	half	free	come	believe
shut	sister	card	other	tried
50	oprah	gift	doing	huge
@50cent	meets	secret	bored	lee
cent	reveal	picked	chat	shooting
wshh	adoption	receive	each	guilty
shutting	oprah's	$1,000	http://tinychat.com	murder
shuts	winfrey	victoria's	tinychat	mass
worldstarhiphop	@oprah	chosen	chatroom	arizona
shuttin	patricia	selected		suspect
	half-sister	drawn		accused (...)
304	**82**	**62**	**47**	**42**

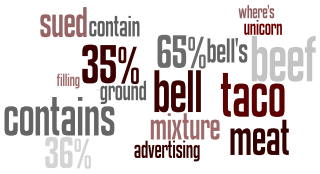
Table 6.8: Most important clusters and magnitude score for 24 January 2011 9-10pm G.M.T.

'cheaper' and **'shower'** clusters are not related to any event at that time. These consist however of words that can be related pairwise to each other (e.g. 'melissa' and 'rivers', 'cheaper' and 'replacement', 'cheaper' and 'originally') in terms of high co-occurrence. In this case, the clustering algorithm's capacity of creating clusters in which not all the pairwise similarities are high fails and adds a *chain* of words, only few with high similarities.

We also observe clusters such as **'free'** and **'come'** where all words are related, but are not uncovering any event. These are likely caused by multiple very similar messages, some automatic. Because these messages are slightly altered, the deduplication step fails to filter them. Words in these messages will have very high NPMI values and thus they would be discovered in the same cluster. Our goal of capturing distributionally similar words is illustrated by the presence in the same cluster of pairs of words (e.g. 'wshh' and 'worldstarhiphop', 'techcrunch' and 'tctv') which very rarely co-occur but are an abbreviation of the other and thus representative of the same event (i.e. users will use the terms interchangeably but rarely at the same time).

Selected events For an in-depth analysis we chose a number of known real-world events that happened during our data collection interval. For objectivity, we used a subset of events extracted from the queries of the TREC Microblog track 2011.[5] This track addresses a realtime search task, where the user wishes to see the most recent relevant information to the query. The organisers also provided 50 queries for this task. While no narrative and description tags are provided, the topic assessor has a clearly defined information need. These queries thus actually represent real-world event keywords.

[5] `http://sites.google.com/site/microblogtrack/`



Query: Taco Bell filling lawsuit
Label: Taco Bell defends its mixture of seasoned meat http://bit.ly/efIzP3
Coherence: 0.38
Magnitude: 73
Date: 25 Jan 2011, 10-11pm

Query: Moscow airport bombing
Label: Suicide bomber kills 35 at Moscow airport http://ind.pn/idWMJj
Coherence: 0.30
Magnitude: 214
Date: 24 Jan 2011, 5-6pm

Query: Kubica crash
Label: Formula 1 driver Robert Kubica injured in rally crash http://ow.ly/3R7IQ
Coherence: 0.35
Magnitude: 140
Date: 6 Feb 2011, 12-1pm

Query: Oprah Winfrey half-sister
Label: Oprah Winfrey has a half-sister. http://bit.ly/i7NNjs
Coherence: 0.25
Magnitude: 43
Date: 24 Jan 2011, 9-10pm

Query: Toyota recall
Label: Toyota recalls nearly 1.7 million vehicles http://lsnlw.com/t/132876715/
Coherence: 0.33
Magnitude: 230
Date: 26 Jan 2011, 6-7am

Query: U.S. Unemployment
Label: Unemployment 9.0% #unemployment #economy
Coherence: 0.22
Magnitude: 108
Date: 4 Feb 2011 2-3pm

Figure 6.4: Original TREC Microblog query, the most relevant tweet, words in the cluster with font size defined by centrality, coherence, magnitude and date of the events.

We discarded single word queries and those in which any word pair had less than 5 occurrences in any hour. We do this because these pairs could have been discarded in our filtering steps which are necessary for adjusting the NPMI bias for rarely occurring words. Most of the TREC events are of lower interest (e.g. 'the release of the Rite') or are general and static (e.g. 'global warming and weather'). If we used the entire Twitter stream as input, recall would be higher.

We present the clusters associated with the 13 discovered events in Figure 6.4 and continued in Appendix A. The cluster was automatically identified as the cluster that contained the query terms and the time interval as the largest value of the sum of co-occurrence frequencies between all pairs of words in the query.

We notice that we can reliably discover all the events indicated by our queries. A qualitative inspection shows that the clusters contain most of the relevant words to that event and its specific time. For example, in the 'Oprah' cluster we see that

the related event is about the revealing that she has a half-sister who was given to adoption.[6] Her name ('Patricia') together with Oprah's surname, hashtag and username are also present. In the 'Kubica' example, we observe that Robert Kubica was a driver that was badly injured in a rally crash in Italy.[7] We notice other words that describe his activity as a Formula 1 driver, like 'f1', '#f1', 'formula' and the team for which he was racing ('Renault'). The centrality measure emphasises correctly the concepts important to the cluster (e.g. 'Oprah, 'Winfrey', 'sister', 'half-sister' or 'Kubica', 'Robert', 'crash', 'injured'). Moreover, our method finds related words even if they are very frequent in the dataset, which a method based on a tf-idf metric [Manning et al., 2008] will discard because of high document frequency (e.g. 'sister' in the 'Oprah' cluster or 'Italy' in the 'Kubica' cluster). We highlight that our event detection method is unsupervised in that it discovers these events with no supervision or manual tuning.

6.3.3.3 Aggregated analysis

The analysis from the previous section was performed when similarities were computed over a restricted time frame i.e. one hour. In this section we apply the same clustering method but using the similarity matrix computed on the entire two months dataset. These experiments aim to confirm our initial hypothesis: word co-occurrences in a static corpus built over a large time interval reflects semantically related concepts. In such cases the temporal effect is watered down.

Cluster quality We have experimented with a number of clusters k from 50 to 500 in increments of 50. We chose to present results with $k = 100$, which obtained best purity results. This is also similar to the recommended setup for the number of topics in topic models based on topic coherence [Stevens et al., 2012]. In order to provide an overview of the type of clusters the method discovers, we asked two independent annotators to judge if and how each cluster is relevant to either an event or topic. The Inter-Annotator agreement (IAA) is 0.67 and Cohen's kappa $\kappa = 0.66$ (significant at $p < .01$). Results presented in Table 6.9 show a shift in the type of cluster content. Many of the clusters now represent general topics, similar to those discovered by topic models.

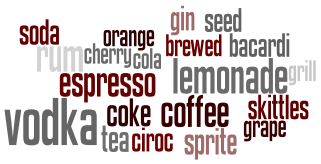
[6] http://www.bbc.co.uk/news/entertainment-arts-12274349

[7] http://news.bbc.co.uk/sport1/hi/motorsport/formula_one/9388940.stm

General topic	Event related	Partial relevance or mix of topics	Not topic or event related
62.5%	10.5%	18.5%	8.5%

Table 6.9: Cluster quality judgements averaged over two independent annotators for aggregated dataset experiments.

Selected clusters We present an in-depth analysis of a selected set of clusters in Figure 6.5. This set aims to be representative of the types of clusters identified in our data. A complete list of clusters presented as a list of the top 10 most central words (using Equation 6.15) is in Appendix B.

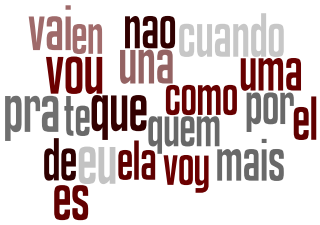


Cluster #28 Label: Drinks

Cluster #31 Label: Words with elongated endings

Cluster #34 Label: Nighttime

Cluster #39 Label: Tech



Cluster #63 Label: Portuguese and Spanish words

Cluster #96 Label: Floods

Figure 6.5: Six sample clusters from the Gardenhose aggregated corpus. Size is proportional to word centrality. The topics are human labelled. Colours are only for clarity purposes.

We first identify clusters in which the words refer to the similar semantic concepts, such as 'Tech' in Cluster #39 or words referring to types of drinks in Cluster #28. Cluster #31 is made up of words that share a particular feature: they all have elongated endings. This is very intuitive on an analysis of the data, as a group of users prefer to write messages using such words. A similar effect leads to clusters of words specific to one particular language as in Cluster #63. Tweets are usually written in a single language resulting in words from that language having high co-occurrence.

The most central words are frequent words from that particular language. These are central to that cluster because they have non-zero co-occurrence scores with most other words in the cluster, having appeared together in tweets. Conversely, English stopwords are not central to any cluster as they occur with words from many clusters, thus 'spreading' their centrality across many clusters. Cluster #34 consists of words which co-occur for a different reason: they all represent words written by people when they leave Twitter at night. Finally, Cluster #96 is considered a more event, rather then topic, related cluster. This is made up of words relating to the Queensland floods in Australia that happened at the start of 2011. Because these spread across a longer time interval and few other tweets mentioned other flood related words, these words were captured within a single cluster. We expect that by further broadening the time interval used for computing the co-occurrence scores, this effect to further diminish.

6.3.4 Austrian Politics experiments

We now present experiments conducted on the Austrian Politics dataset. This has some very distinctive proprieties compared to the Gardenhose dataset. Most importantly, it is considerably smaller volume with data over the entire year being approximately the same volume as 4-5 hours of the Gardenhose dataset. Secondly, the data is from political users who discuss current affairs and thus is more event focused. As highlighted in the quantitative analysis from the previous experiments, a few parameters of our method need to be adjusted. First, we aggregate our data on a daily basis, rather then an hourly one in order to have a larger underlying set of data for co-occurrence computation. Using similar quantitative experiments to those in Section 6.3.3.1, we have determined the number of clusters $k = 30$. After building the nearest-neighbour graph and keeping only the largest connected component we have found an average of only 567 words are 'active' in a given day. Informed by the previous set of experiments, we set the number of clusters to 15 times less than the number of words to be clustered.

6.3.4.1 Daily analysis

We analyse the results obtained on a randomly selected day of our corpus, 31 May 2013. We have used the coherence threshold of 0.1 in order to remove the clusters with low coherence. On a manual inspection these mostly contained conversational and frequent words as well as words in English, a language that is not prevalent in

our dataset which is predominantly in German. Using this threshold we obtain on average 7 coherent clusters in a day. We note this threshold is lower than the 0.15 we used for the Gardenhose experiments. Because the underlying dataset is small, the level of sparsity of the underlying similarity matrix is higher and clusters with higher coherence are harder to find. All the coherent clusters for this hour, with the words sorted by their centrality, are presented in Table 6.10.

istanbul	adria	schau	auto	paar	#gntm	gedacht	vielen
video	alpe	n	interessant	letzten	#femen	weniger	#ff
#occupygezi	hypo	gleich	halt	andere	next	millionen	dank
police	verkauft	ma	vs	tweets	interview	deutschland	denen
polizei	euro	#heuteshow	einer	seine	gestern	menschen	folgen
park	bank	twittern	richtig	#wernerfaymann	find	deutsche	leute
#direngezipark	#hypo	sagen		faymann		leben	anderen
offenbar	mio	#domian		stunden		neuer	
rt	zeitung	kommt		minuten		als	
#blockupy	kaufen			herr		platz	
#istanbul	kleine			meinem		lassen	
#wien	austria			tweet		werde	
today	idee			deine			
folgt	geht			jahre			
gegen				ein			
vor				alles			
				guter			
				thema			
0.12	**0.17**	**0.10**	**0.12**	**0.10**	**0.13**	**0.11**	**0.11**

Table 6.10: Coherent clusters and coherence scores for 31 May 2013.

We observe that some clusters are event related and provide the essential information for uncovering the underlying event. For example, cluster **'istanbul'** contains words referring to the 'Occupy Gezi' movement[8] that started on 29 May 2013 in Turkey, while cluster **'adria'** refers to an economic event which involved the Hypo-Alpe-Adria bank.[9] Cluster **'#gntm'** is about a popular TV show while **'paar'** involves the Austrian chancellor Werner Faymann. However, the other clusters do not uncover events. For example, cluster **'vielen'** contains words in the context of 'follow Friday' ('#ff'), messages which users send on Fridays in order to get more followers. We also observe many very frequent terms which were not filtered as in the Gardenhose experiments. We conclude that while the key events are identified, further filtering for words is needed to remove noise. These inferior results are caused by the small volume of data (∼5000 tweets/day), which do not lead to an accurate computation of the co-occurrence values and by simply the reduced number of events that exist in the underlying dataset.

[8] http://en.wikipedia.org/wiki/2013_protests_in_Turkey

[9] http://www.reuters.com/article/2013/05/31/austria-hypoalpeadria-idUSL5N0EC17220130531

6.3.4.2 Aggregated analysis

In this section we perform experiments with co-occurrences computed on the aggregated dataset in a similar way as in the Gardenhose experiments from Section 6.3.3.3. Although the dataset spans a large time interval (one full year), it is still lower volume (∼300 times) compared to the Gardenhose dataset. Experiments were conducted with $k = 200$ clusters using spectral clustering.

Cluster quality In order to asses cluster quality, SORA[10] subjectively assessed the coherence of each cluster due to their expertise and domain knowledge in Austrian politics and affairs. The coherence scores were categorised as being: 'Not topic or event related', 'Partial relevance or mix of topics' and 'Event related' or 'General topic'. Results are shown in Table 6.11.

General topic	**Event related**	**Partial relevance or mix of topics**	**Not topic or event related**
8%	60.5%	16.5%	15%

Table 6.11: Cluster quality judgements averaged over two independent annotators for the aggregated dataset experiments.

The results are very different from the aggregate results on the Gardenhose dataset from Table 6.9. Whereas the 'General topic' category was the most prevalent earlier, here it is the least prevalent. Conversely, the 'Event related' clusters are the most frequent in this set of clusters. This can be explained by the following factors: dataset size and the type of messages. Messages are authored by users that mainly discuss current affairs and news, with a restricted number of conversational or general purpose messages. This is expected to heavily influence the co-occurrence statistics. Most words rarely appear in their *natural* context, but only in event related ones, pushing the co-occurrences away from semantically similar concepts. The smaller volume of the dataset is likely to play an important role as well, as many words only occur during a related event which lasts generally a few days.

Selected clusters A detailed view of a few selected clusters obtained using the aggregate dataset is presented in Figure 6.6.

We first identify the clusters related to specific events from the one year time interval. Clusters #70 and #92 present events of international importance such as

[10]The institute that also helped assemble the dataset.

Cluster #30 Label: Elections for the Austrian student union

Cluster #70 Label: Papal election

Cluster #92 Label: Horse meat scandal

Cluster #196 Label: Southern German/Austrian slang words

Figure 6.6: Four sample clusters from the Austrian Politics dataset. Size is proportional to word centrality. The clusters are manually labelled. Colours are only for clarity purposes.

the Papal election[11] and the horse meat scandal.[12] By analysing the words from Cluster # 70 we can identify words related to the election ('#papstwahl' - papal election, '#pontifexit', '#konklave' - conclave), to the location ('#vatikan - Vatican,'#sixtinische' - Sistine (Chapel), 'petersplatz' - St. Peter's Place), the previous Pope ('benedikt') and the newly elected Pope, the Argentinian Franciscan archbishop of Buenos Aires, Jorge Mario Bergoglio as Pope Francis I ('jorge', '#bergoglio', 'argentiner') Cluster #92 identifies the key terms as being related to meat and its origin ('pferde' - horse, 'rind' - beef, 'fleisch' - meat). Also relevant are identified a supermarket that was alleged to have commercialised horse meat ('lidl') as well as the products that contained it ('tortelloni'). Cluster #30 reflects a local Austrian event: the elec-

[11] http://en.wikipedia.org/wiki/Papal_conclave,_2013

[12] http://www.theguardian.com/uk/2013/may/10/horsemeat-scandal-timeline-investigation

tions for the president ('vorsitzender') of the Austrian Student Union (Österreichische HochschülerInnenschaft – ÖH or OEH) in June 2013.[13] We identify the key hashtags associated to these elections ('#oeh13', '#oehwahl2013'), the parties involved ('gras', 'ag', ''diefest') as well as the winner of the election ('#kraushofer'). Cluster #196 is labeled as a general topic. This contains words from Southern Germany and Austrian slang and are used only by a subset of users in their messages.

6.3.5 Application to summarisation

In order to label the clusters, we extract relevant tweets by using the words in a cluster and their centrality score. Given a cluster, we compute for each tweet a weighted sum of its tokens, where the weight is the word's centrality score in the cluster:

$$S_C = \frac{\sum_{w \in C} \mathrm{Ce}_w}{|C|} \tag{6.17}$$

In order to remove short tweets that are not suited for our purpose we keep only tweets with more than 3 tokens. Becker et al. [2011a] shows that, even though using a different measure of similarity and centrality, when finding representative tweets in a collection, weighting by the centrality of words performs well. Examples of relevant tweets for their cluster are shown as cluster labels in Figure 6.4. We find that the most representative tweet to a cluster is a well written piece of text that describes that event succinctly.

In order to evaluate label quality we perform a human experiment. This experiment also indirectly evaluates the quality of the words within a cluster. The annotators were presented for each of the previously identified 13 TREC microblog events (see Section 6.3.3.2) with a URL giving background information about the event and the following tweets: 3 random tweets for each 'highly relevant', 'relevant' and 'not relevant' category as judged as part of the 2011 TREC Microblog task evaluation and the top 3 most relevant tweets as found by our method using the same underlying data. The question they had to answer was: 'On a scale from 1-5, how well each tweet is suitable as a label for that event': 1 - tweet is not relevant; 2 - tweet is somewhat relevant; 3 - tweet is relevant but not suitable as a label/summary; 4 - tweet is relevant and can be used as a summary, but contains too little information or extra unrelated words; 5 - tweet is relevant and can be used as a summary'. Each annotator judged all the 13 events. The average variance across judges is 0.38. IAA

[13] `http://diepresse.com/home/bildung/universitaet/oehwahl/1424287/Florian-Kraushofer-ist-neuer-OHChef`

is computed as the average Spearman's ρ between the scores given by the annotator and the average ratings given by all other annotators. The average IAA across all topics was 0.875. The results presented in Table 6.12 show that our method outputs the best labels for events, with a very high average score. This shows that the words in our clusters identify the correct events and provide a good match for the queries.

Method	**Average score**
Our method	4.33
TREC judged 'Very relevant'	3.61
TREC judged 'Relevant'	3.13
TREC judged 'Not relevant'	1.53

Table 6.12: Relevance judgement results.

6.4 Temporal experiments

Given the clusters and events uncovered in the previous section we can study their behaviour over time with the help of our evolutionary clustering algorithm. We highlight that cluster correspondence is performed automatically by our evolutionary clustering algorithm without needing to solve matching for cluster identifiability. In addition to a lower computational cost, this also results in more consistent longitudinal clusters.

Results for two events are shown in Figure 6.7 with emphasis on the content drift and the evolution in magnitude over time.

In the 'Oprah sister' event, we observe some buzz before the announcement because she advertised that she will reveal a 'family secret' without further details. A few hours before the announcement, people found out that she would reveal her half-sister. The peak in magnitude is when the announcement was made on the morning show in the U.S. (time is G.M.T.) and there is another smaller peak when her afternoon show aired. By then, people were already talking about her sister's name ('Patricia') and that she was given up for adoption.

For the 'Kubica crash' event we notice a more abrupt trend. When the rally crash happened, there were many users reporting it, and afterwards the magnitude decays quickly. In the first hour of the crash, people were reporting the news without many details or comments ('f1', 'rally', 'crash'). After a few hours, the number of words relating to the event increased as more news and different comments about the driver are made. After 12 hours from the event, we notice a drift in content. Many people are talking about the surgery the driver had and are wishing him a 'speedy recovery' while others still mention the rally crash.

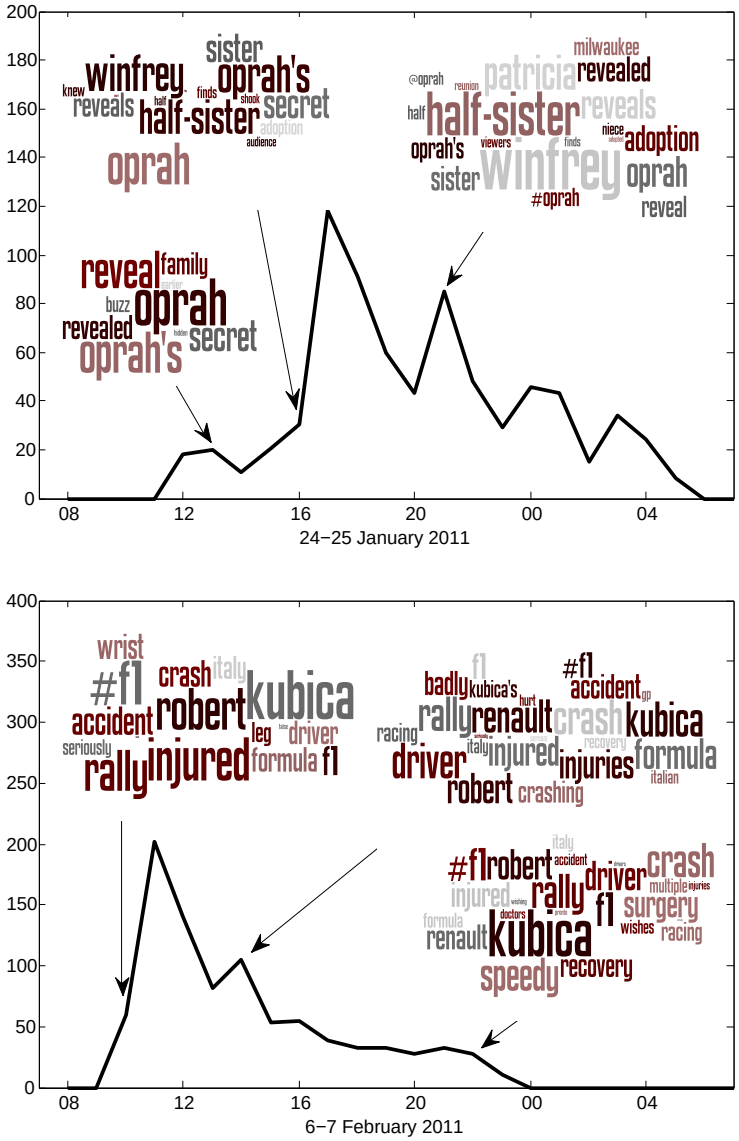


Figure 6.7: Cluster evolution in magnitude and words over time.

6.5 Conclusion

This chapter introduced methods that exploit the temporal changes in word proprieties in order to detect the events that generated them. The method of event identification is based on computing word co-occurrences within a time window. Using this data representation, we proposed a hard spectral clustering method to identify word clusters. By experimenting with two very different datasets (in size and language) we have identified that word co-occurrence clustering on a small time interval of OSN data can uncover events (timely newsworthy happenings) in that dataset. By increasing the time interval and data volume, this effect is reduced and words are no

longer identified with words relating to the same event, but broadly to semantically similar classes ('topics'). However, experiments on a smaller dataset highlighted the importance of a high data volume in order to dampen the influence of event related words. On this dataset, focused on reporting and discussing current affairs, the clusters discovered using one year of data mainly identify events. We also studied a series of outcomes of the clustering method, such as weighting the words by their centrality to determine the important concepts of the cluster, cluster coherence as a way of filtering uninteresting clusters and identifying the most relevant tweets for a cluster as a way of summarising the content.

Further, we have shown how to track events over time, incorporating temporal smoothness in an evolutionary spectral clustering method. Our methods showed that abrupt changes in word distributions in OSN data are triggered by real-world events whilst being efficient and robust. Similarly to Chapter 4, we have only considered a smooth temporal effect, by which the evolution of text is smooth across time and influenced by the previous time step. More complex temporal modelling techniques presented in the Chapter 5 can be combined with the current approach e.g. to identify periodic events and their content.

Chapter 7

Conclusions

This thesis studied the impact of time on social media data. The work undertaken in this thesis provides an extensive evidence that the properties of social media data change conditioned on time. As people switched many activities to the online medium, it is expected that social media should contain useful information about their real world behaviours as well about real events. Based on the significant correlations discovered and strong empirical accuracy of social media data to real world indicators and the correlation of clusters extracted from data to real events, we can conclude that social web data provides a timely reflection of the real world. Time has proven to play essential role in the type of information contained and including time explicitly into modelling is highly beneficial.

Throughout this thesis we developed several ways of modelling time using machine learning methods. These used temporal social media data in order to discover correlations to real world indicators, for making forecasts and for event emergence and tracking. We have used data from two different types of OSNs, with a strong emphasis on text, which provide temporal streams of short documents and user behaviours. Time was modelled in two different ways. First, in both a supervised and an unsupervised setting, we have assumed a smooth evolution of data with time. Although this is an intuitive and effective use of temporal information, more complex temporal dependencies also exist in the data which have not been previously explored in natural language processing research. We have developed a supervised method that focuses on identifying periodicities from both word and user behaviour data. Our methods had to deal with practical challenges, such as large volume or non-standard input, as well as algorithmic challenges, such as reasoning over large collections of messages where few are relevant to our goals.

Our first major contributions are temporal models for forecasting. We develop two different methods, each modelling different temporal effects. First, we develop super-

vised regression models which correlate text features extracted from social media data with real world indicators, namely political party voting intention. We introduce a bilinear regression model with sparse regularisation which uncovers a sparse weighted set of both user and text features used in a multi-task prediction setting. Further, an online algorithm is developed to work effectively in a real world streaming scenario, where voting intentions are predicted whenever new data is available. We introduce in this model a temporal smoothness constraint, where older data is gradually forgotten in the detriment of new data. Experiments on two different use cases show very good predictive results for the forecasting task, outperforming established methods. Incorporating temporality also proved beneficial, showing that performance gains resulting from simple assumptions on the temporal dynamics.

Our second forecasting model deals with identifying and categorising complex periodic patterns in data. We use Gaussian Process regression to model time series of hashtags frequencies in Twitter and user behaviour in LBSNs. We introduce a kernel which can effectively model a specific behaviour of frequencies on social media: *spiking* patterns characterised by an abrupt periodic rise followed by very low frequency intervals. Experiments show excellent predictive performance in an extrapolation setting, with predictions being made weeks in advance. A classification is automatically performed by the kernel selection, uncovering the periodic and non-periodic time series. Further, we incorporate the hashtag and user behaviour frequency forecasts in downstream applications: text classification and user beheviour prediction. We demonstrate the power of the temporal pattern information by obtaining improvements over standard methods even with a straightforward integration of the temporal information.

Another major contribution is an unsupervised method of event discovery and tracking. Newsworthy events are reflected by multiple posts on social media and will likely influence word properties. Here, we have studied the temporal evolution of word co-occurrences. We have developed an efficient spectral clustering method which groups words based on their co-occurrence within a time window. Human and automatic evaluation showed discovered that in restricted time windows, co-occurrence clusters are highly indicative of events. However, with the expansion of the time window, words tend to co-occur more with semantically related concepts, overcoming the effects of short lived events.

At the foundation of all our methods and experiments stands an emphasis on replicability, generality and independence from external resources. All our methods can be directly applied with any other textual data, provided we can extract textual

and temporal features, as we have not used features specific to the Twitter social network. We have also proven generality across use cases by performing experiments on data from a different language (German cf. English). Throughout the thesis, we have not used any data or resources external to the raw social media input which makes methods portable to new applications. The only exception is the political polling data used as supervision in the voting intention experiments. We note that the models tested on the polling data can be applied in order to discover correlations with other real world time series. By modelling periodicities on Twitter hashtags, we have discovered that similar patterns also exist in the case of user behaviour, showing wider applicability. For the unsupervised clustering algorithm, different experimental setups were considered which uncovered the effects of different types and volumes of input data. Social media data is of high volume and subject to privacy constraints, which makes dataset distribution and experiment replicability an issue. However, feature representations and data collection code are provided openly together with guidelines in the processing steps used for each experiment.

Another contribution of this thesis is the development of a text pre-processing pipeline tailored for social media data. Our pre-processing pipeline contains all operations needed in order to transform a raw social media dataset to feature representations: tokenisation, language detection, data deduplication, etc.

7.1 Further research

Methods developed as part of this thesis can be more generally used in other applications and fields of research. We mention some directions for further research:

- The bilinear model introduced in Section 4.3.3 can be further extended in multiple ways by modifying either the task or how the regularisation is performed. First, we can apply a similar model to a classification task, where the response is the presence or absence of an event (e.g. social unrest). Further, the initial model, even using the same data from our experiments, can be correlated to other time series from different domains (e.g. economics). The multi-task learning scenario can be extended in order to incorporate user specific information (e.g. demographics, location) as well as voting intentions conditioned on these factors. Preliminary results are presented in [Preoţiuc-Pietro et al., 2014]. From a regularisation perspective, we can use semantic information about words (e.g. word clusters) to build hierarchical or overlapping groups for regularisation. This biases groups of semantically related words to be selected simultaneously,

leading to more interpretable results. Other deeper linguistic information, such as named entities or ontologies, can also be used as features in order to gain a better understanding of the results. Collaborations with researchers from the social sciences can aid the interpretation of the models. Throughout our work, we have assumed the bilinear components represent user and word features. However, other feature sets, such as news and news sources, can be used with no change in algorithms.

- Complex patterns, such as the periodic ones identified in Chapter 5, can be further combined with models assuming temporal smoothness or integrated in other methods (e.g. as a prior in topic models). Topic models (Section 4.1.1) aiming to model time make simplistic assumptions on the temporal dependencies. More complex dynamics, such as described by a mixture of radial basis functions or periodic functions, can be used in conjunction to models such as LDA (Latent Dirichlet Allocation) [Blei et al., 2003] or DMR (Dirichlet Multinomial Regression) [Mimno and McCallum, 2008].

- The text classification experiments in Section 5.2.7 use the temporal forecast value as a prior in a Naïve Bayes classifier. This motivates future work to use this information in discriminative classifiers, thus avoiding the need for the Naïve Bayes decomposition. The GP modelling framework can accommodate classification (through the logistic likelihood), although scaling issues arise when using a large number of features or output classes.

- Although this thesis focuses on studying the venue type information in LBSN check-ins, the underlying data is much richer. Multiple modalities exist, such as text or images, and combining them into a joint model is a promising research area. Temporal check-in patterns, as well as category and text information, can also be used to identify specific characteristics of cities or neighbourhoods [Preoţiuc-Pietro et al., 2013a]. Studying neighbourhood and city similarities using temporal and textual features is a less explored area.

- Word co-occurrence scores computed over restricted time intervals can aim further downstream applications. Information retrieval techniques can use temporal co-occurrence information in order to provide temporally aware responses to users. The clusters discovered by our method can be used for dimensionality reduction of the word feature space. The Gardenhose clusters from Appendix B

have been used successfully in [Lampos et al., 2014] for predicting and characterising user impact on Twitter using user profile features combined with cluster/topic features. The clusters provide extra interpretability on the topics which determine higher user impact.

- The open-source data pre-processing tool presented in Chapter 3 can be further extended with other components. Some suggestions are presented in Section 3.1.8, such as text normaliser, Part-of-Speech tagger or an entity recogniser. Extending this tool will allow for easier access to social media data and NLP techniques. Researchers from other disciplines, such as the Arts & Humanities, should be encouraged to engage in collaborative research projects.

7.2 Summary

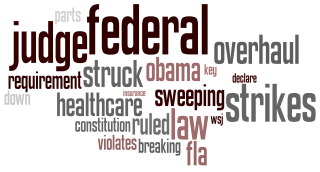
In summary, this thesis highlights the importance of modelling time in applications using social media data. Both supervised and unsupervised machine learning methods are introduced for modelling either smooth temporal relationships in data or long term complex temporal patterns. Together, these methods perform event emergence and tracking, discovery of correlations to real world indicators and forecasting of voting intentions, word frequency and user behaviour. Our methods allow further integration into downstream tasks, such as classification, as well as providing support for human analysis of social media data.

Appendix A

Event related clusters

Query: Cuomo budget cuts
Label: Official: NY's Cuomo proposing painful budget cuts http://feeds.foxnews.com/...
Coherence: 0.27
Magnitude: 662
Date: 1 Feb 2011 4-5pm

Query: Egyptian curfew
Label: Military issues curfew in Cairo, Alexandria and Suez
Coherence: 0.24
Magnitude: 132
Date: 28 Jan 2011 3-4pm



Query: healthcare law unconstitutional
Label: Judge strikes down Obama healthcare law
Coherence: 0.24
Magnitude: 125
Date: 31 Jan 2011 8-9pm

Query: Keith Olbermann new job
Label: Keith Olbermann heads to Current TV http://lat.ms/dHxYnq
Coherence: 0.24
Magnitude: 198
Date: 6 February 2011 12-1pm

Query: protests in Jordan
Label: Jordan's King Abdullah dismisses government http://lat.ms/gMa9wt
Coherence: 0.36
Magnitude: 398
Date: 1 Feb 2011 2-3pm

Query: Kucinich olive pit lawsuit
Label: Dennis Kucinich Sues Congressional Cafeteria Over Olive Pit http://gaw.kr/eiuDIH
Coherence: 0.25
Magnitude: 128
26 Jan 2011, 8-9pm

Figure A.1: Original TREC Microblog query, the most relevant tweet, words in the cluster with font size defined by centrality, coherence, magnitude and date of the events (continued on next page).

Query: White Stripes breakup
Label: Hmm White stripes split
Coherence: 0.20
Magnitude: 439
Date: 2 Feb 2011, 6-7pm

Figure A.1: Original TREC Microblog query, the most relevant tweet, words in the cluster with font size defined by centrality, coherence, magnitude and date of the events.

Appendix B

Aggregated clusters

No.	Top words by centrality
1	tna , #wwe , #wrestling , whether , wcw , facebooking , hogan , wrestlers , speaking
2	yu , floored , mateee , usher , wens , urs , #confessions , ight , #movingon
3	blvd , walt , angeles , nv , bellagio , wynn , brazilian , tenor , santa
4	romeo , sorrows , shakespeare , upon , playboy , aguilera , hill , vaccine , grave
5	hooooo , hooo , hellll , helll , woooo , whoooo , wooooo , whooo , hoo
6	40's , 30's , 50's , 20's , 60's , freezing , warmer , 60s , upper
7	fi , seh , mek , gyal , sef , suh , weh , tek , inna
8	wine , tasting , wines , pinot , #beer , winery , brewing , gardening , foods
9	disorder , outward , aristotle , orchestra , anxiety , napoleon , rohn , inward , sexual
10	roasted , spinach , tomato , potatoes , chicken , salad , soup , cheese , onions
11	contributors , null , favorited , a3 , hashtags , href , rel , geo , a2
12	nurse , rn , registered , bedroom , clinical , #news , estate , #hospital , healthcare
13	fucka , mutha , shoulda , coulda , woulda , anotha , stayed , brotha , fucker
14	#online , #ebook , #kindle , #tv , #film , #ebooks , #read , #media , #book
15	disagree , wholeheartedly , anyone , twit , least , pic , else , at , southerners
16	kinda , basically , shunned , folkestone , shunning , beefed , leas , 1902 , blacklisted
17	bercow , reprimanded , #conservatives , theroux , devey , corner , #sp11 , wife , boothroyd
18	sooner , brrr , brrrr , 909 , txt , brr , #h50 , five-0 , evah
19	zoom , canon , lens , intel , nikon , optical , digital , camera , lcd
20	#cricket , cricket , sachin , #nba , #steelers , federer , #tennis , tennis , #icd
21	hustle , #hard , brilliant , #hardwork , #grind , #dedication , stelling , tamsin , lineker
22	anniversary , cena , orton , wrestlemania , undertaker , miz , hhh , smackdown , wrestler
23	#diet , #health , remedies , #natural , #body , #exercise , acne , loss , #weightloss
24	psalm , christ , #nhl , proverbs , unto , salvation , psalms , lord , righteousness
25	astrology , zodiac , aries , libra , aquarius , pisces , taurus , virgo , horoscopes
26	rick , swizz , kanye , jay-z , biggie , t-pain , nicki , hilson , wale
27	#10oclocklive , #bbcqt , #ihate , #blah , #bbctw , #ilove , huq , portillo , wagwan
28	lemonade , rum , coffee , espresso , coke , sprite , tea , soda , skittles
29	blackburn , manchester , spurs , sunderland , redknapp , bolton , #epl , #thfc , milan
30	meeeee , youuuuu , doooo , dooooo , shucks , meeeeee , goooo , tooooo , yoooou

No.	Top words by centrality
31	dayyy , theee , goood , alll , youuu , loveee , folllow , birthdayyy , happpy
32	rican , puerto , dulce , leche , cabo , cinco , verde , calle , pollo
33	terminal , fri , wed , #myfitnesspal , march , international , earthquake , tues , ord
34	tweeties , sweetdreams , gtg , #goodnight , goodnite , #sweetdreams , gnite , g'night , beddy
35	roadworks , northbound , southbound , junctions , #job , #jobs , eastbound , westbound , analyst
36	#sex , #hardcore , #fuck , #boobs , #tits , #amateur , #sexy , #milf , #ass
37	nak , gw , gue , kalo , itu , aku , aja , ini , klo
38	necklace , sterling , pendant , 14k , silver , bracelet , bead , jewelry , #handmade
39	nintendo , #xbox360 , #ps3 , #gaming , #pc , #apps , ps3 , #apple , xbox
40	exceeding , barney's , himym , moonwalking , sky's , mix-up , forsee , skys , ladies
41	golly , fxck , derp , bbz , herp , looool , looooool , dyin , looooooooool
42	cheek , smiles , grins , gently , tummy , leans , giggles , kisses , stomach
43	eyebrows , beard , shave , nail , waxing , infection , eyebrow , pierced , piercing
44	ankle , terrier , jeans , shorts , muscles , dogs , spider , yorkie , knee
45	ewwww , dovey , lovey , ewww , ew , ewwwww , ewwwwww , eww , grossed
46	#marvel , #african , #xmen , #spiderman , #american , #trailer , #captainamerica , #super8 , #deadisland
47	republican , gop , police , arrested , voters , robbery , democrats , presidential , charged
48	fingering , bbw , khalifa , bdsm , dogg , wiz , lesbians , feat , milfs
49	e.i , t.i , tae , shi , walla , shay , h.i , ee , n.i
50	#webdesign , #freelance , php , css , sql , mysql , wordpress , #php , #seo
51	#celebrity , #celebs , #fame , #rock , #guitar , #famous , #celeb , #peace , #musicmonday
52	#500aday , #tfb , #instantfollowback , #ifollowback , #instantfollow , #followback , #teamautofollow , #autofollow , #follow4follow
53	stargazing , #bbc2 , #bbcstargazing , #lakeside , #bdo , wolfie , bdo , dara , bbc2
54	potter , #tvd , harry , elena , kate , portman , pattinson , hermione , kristen
55	tanya , bardsley , bramble , snodgrass , yummers , safc , asamoah , #shocking , equalise
56	inventory , coupon , toyota , mileage , sedan , nissan , adde , jeep , 2002
57	wacko , jacko , gypsys , #mybigfatgypsywedding , caravans , #bigfatgypsyweddings , #bigfatgypsywedding , pikeys , appleby
58	withh , justt , forr , shee , myy , likee , thatt , andd , itss
59	frack , frick , sheik , rearing , #fuckthis , strung , corporates , ugly , ughhh
60	#energy , #eco , #green , efficient , #recycle , #solar , nature , #home , #relaxation
61	#tunisia , #iran , #israel , #palestine , tunisia , arab , #jan25 , iran , protests
62	affiliate , affiliates , converting , earn , 75% , epc , recurring , commission , #education
63	vou , pra , que , es , una , vai , cuando , el , nao
64	20s , 40s , 50s , highs , 10s , utc , temps , rounds , gusts
65	shore , sammi , deena , soulja , waka , vinny , flocka , liams , aunts
66	9:00 , nws , 6:00 , 10:00 , 7:00 , 12:00 , p.m. , 4:00 , 5:00
67	anche , io , santiago , rio , bene , brazil , janeiro , colombia , ciao
68	reblog , texts , reblogging , fb , reblogged , deactivated , bbms , msgs , messages
69	iglesias , enrique , rome , angelina , nicole , erykah , scherzinger , jr , seth
70	niall , zayn , horan , jls , avoi , n-dubz , marvin , merrygold , marry

No.	Top words by centrality
71	je , ook , niet , wel , maar , een , nog , op , het
72	#aries , #leo , #aquarius , #gemini , cultural , #pisces , awarded , #capricorn , attraction
73	30mins , 2hrs , 15mins , 3hrs , 20mins , 45mins , 5hrs , 4hrs , 1hour
74	syndrome , patients , ski , dental , chronic , diagnosed , #dog , prostate , cardio
75	winnie , grin , stares , smiled , fist , sits , bollocks , moaning , twat
76	pips , eurusd , #forex , volume , 12% , gbpusd , 13% , 4% , reduced
77	243 , #twalue , 83 , 251 , 221 , 203 , 216 , 89 , 227
78	salman , srk , #bollywood , kapoor , rukh , kaif , karan , kansas , martial
79	avenged , mentally , deepest , a7x , physically , sexually , confessions , courage , cease
80	exceeded , urls , further , error , limit , dary , wait , rate , legen
81	jl , jalan , raya , barat , cok , kota , indonesia , hic , bsd
82	ping , likes , vista , punto , fiat , epcot , bizjournals , com , logo
83	bmx , bicycle , bikes , fitness , stroller , guitar , curves , drums , pedals
84	tweet , dedicating , 1000th , #teambieber , 000th , 2000th , #beliebers , 3000th , beliebers
85	continue , tristan , cracking , creaking , 20mph , pelted , rpc , 60mph , obscenities
86	carew , #stoke , delap , shawcross , #fulham , #ffc , outside , whitehead , bestie
87	kuala , bangkok , malaysia , malaysian , jaya , plaza , phuket , thailand , rama
88	wish , luck , proceeding , wished , appealed , could , good , 11.11 , wishes
89	#arsenal , #lfc , fabregas , cesc , nasri , torres , liverpool , wenger , chelsea
90	i‘m , it‘s , can‘t , you‘re , i‘ll , i‘ve , y.i , posner , u.i
91	#6nations , #wales , #6n , #bbcsixnations , cueto , tindall , #heartbeats , #whoyouare , #eng
92	olivia , remain , disagreeing , old , agreeing , heartily , dull , silent , focus
93	ako , ko , naman , lang , sa , kasi , ang , nga , yun
94	humidity , barometer , gust , winds , hpa , temperature , kt , #weather , 0.0
95	7.30 , 8.30 , 6.30 , 30am , ago , weeks , 5.30 , 9.30 , 10.30
96	floods , queensland , avenue , qld , resort , hotel , resorts , yelp , beach
97	avatar , #england , airbender , #scotland , #avatar , ospreys , toulon , #hcup , twickenham
98	dollars , narcolepsy , #millionpounddrop , indycar , sleepwalking , #indycar , marple , #mpd , coyne
99	bhi , nahi , aur , kya , kuch , hain , kar , mein , aap
100	huddersfield , swindon , #htafc , terriers , robins , #stfc , #suarez , #torres , aldershot

Table B.1: All clusters computed over a two months aggregated Gardenhose dataset represented by their top 10 most central words.

Bibliography

Faiz Al-Khayyal and James Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983.

Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics*, IWCS, pages 13–22, 2013.

James Allan. *Topic Detection and Tracking: Event-based information organization*. Kluwer Academic Publishers, 2002.

Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. In *Proceedings of the 2008 8th IEEE International Conference on Data Mining*, ICDM, pages 3–12, 2008.

Loulwah AlSumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi. Topic significance ranking of LDA generative models. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD, 2009.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3): 195–266, 2012.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-Task feature learning. In *Advances in Neural Information Processing Systems*, NIPS, 2007.

Sitaram Asur and Bernardo A. Huberman. Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, WI-IAT, pages 492–499, 2010.

Sitaram Asur, Bernardo Huberman, Gábor Szabó, and Chunyan Wang. Trends in social media: Persistence and decay. *CoRR*, 2011.

Francis R. Bach. Bolasso: model consistent LASSO estimation through the bootstrap. In *Proceedings of the 25th International Conference on Machine Learning*, ICML, pages 33–40, 2008.

Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web*, WWW, pages 61–70, 2010.

Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer: quantifying influence on Twitter. In *Proceedings of the 4th International Conference on Web Search and Data Mining*, WSDM, pages 65–74, 2011.

Timothy Baldwin and Marco Lui. Language identification: the long and the short of the matter. In *Proceedings of the 2010 annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 229–237, 2010.

David Bamman, Brendan O'Connor, and Noah Smith. Learning latent personas of film characters. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics*, ACL, 2013.

Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, ACL, pages 26–33, 2001.

Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding algorithm for linear inverse problems. *IAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Hilla Becker, Mor Naaman, and Luis Gravano. Selecting quality Twitter content for events. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, ICWSM, 2011a.

Hilla Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: real-world event identification on Twitter. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, ICWSM, 2011b.

David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML, pages 113–120, 2006.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

Johan Bollen, Huina Mao, and Xiao-Jun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

Léon Bottou. *Stochastic Learning*. Lecture Notes in Artificial Intelligence, LNAI 3176. Springer Verlag, 2004.

Léon Bottou. Large-scale Machine Learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics*, COMPSTAT, pages 177–187, 2010.

Gerlof Bouma. Normalized (Pointwise) Mutual Information in collocation extraction. In *Proceedings of German Society for Computational Linguistics & Language Technology Conference*, pages 31–40, 2009.

Danah Boyd, Scott Golder, and Gilad Lotan. Tweet, tweet, retweet: conversational aspects of retweeting on Twitter. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, HICSS, pages 1–10, 2010.

Thorsten Brants, Francine Chen, and Ayman Farahat. A system for new event detection. In *Proceedings of the 26th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, pages 330–337, 2003.

Dirk Brockmann, Lars Hufnagel, and Theo Geisel. The scaling laws of human travel. *Nature*, (7075):462–465, 2006.

Samuel Brody and Nicholas Diakopoulos. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!: using word lengthening to detect sentiment in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 562–570, 2011.

Joaquin Quiñonero Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian Process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

Elizabeth Cano, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie. Making sense of microposts (#MSM2013) concept extraction challenge. In *Concept Extraction Challenge, Workshop on 'Making Sense of Microposts', 22nd International World Wide Web Conference*, WWW, 2013.

Simon Carter, Wouter Weerkamp, and Manos Tsagkias. Microblog language identification: overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation*, 47(1):195–215, 2013.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*, WWW, pages 675–684, 2011.

William Cavnar and John Trenkle. N-Gram-Based text categorization. In *Proceedings of 3rd annual Symposium on Document Analysis and Information Retrieval*, SDAIR, pages 161–175, 1994.

Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining*, SIGKDD, pages 554–560, 2006.

Jonathan Chang and Eric Sun. Location3: How users share and respond to Location-Based data on social networking sites. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, ICWSM, 2011.

Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM, pages 759–768, 2010.

Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z. Sui. Exploring millions of footprints in location sharing services. In *Proceedings of the 5th International Conference on Weblogs and Social Media*, ICWSM, 2011.

Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM International Conference on Knowledge Discovery and Data Mining*, SIGKDD, pages 153–162, 2007.

Alvin Chin and Daqing Zhang. *Mobile Social Networking: An innovative approach.* Springer Verlag, 2013.

Hyunyoung Choi and Hal Varian. Predicting the Present with Google Trends. Technical report, 2011.

Fan Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.

Kenneth Ward Church and Patrick Hanks. Word association norms, Mutual Information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

Trevor Cohn and Lucia Specia. Modelling Annotator Bias with multi-task Gaussian Processes: an application to machine translation quality estimation. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics*, ACL, 2013.

Aron Culotta. Detecting influenza outbreaks by analyzing Twitter messages. *arXiv:1007.4748*, 2010.

James Richard Curran. *From distributional to semantic similarity.* PhD thesis, University of Edinburgh, 2004.

Hal Daume III. Markov random topic fields. In *Proceedings of the Joint Conference of the 47th annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*, ACL-IJCNLP, pages 293–296, 2009.

Leon Derczynski, Diana Maynard, Niraj Aswani, and Kalina Bontcheva. Microblog-Genre noise and impact on semantic annotation accuracy. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT, pages 21–30, 2013a.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. Twitter part-of-speech tagging for all: overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP, 2013b.

Son Doan, Bao-Khanh Ho Vo, and Nigel Collier. An analysis of Twitter messages in the 2011 Tohoku earthquake. In *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 58–66, 2011.

Peter Sheridan Dodds, Kameron Decker Harris, Isabel M. Kloumann, Catherine A. Bliss, and Christopher M. Danforth. Temporal patterns of happiness and information in a global social network: Hedonometrics and Twitter. *CoRR*, abs/1101.5120, 2011.

Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

Nicolas Durrande, James Hensman, Magnus Rattray, and Neil Lawrence. Gaussian Process models for periodicity detection. In *Submitted to JRSSb, http://arxiv.org/abs/1303.7090*, 2013.

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the International Conference on Machine Learning*, ICML, 2013.

Nathan Eagle and Alex Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

Jacob Eisenstein. What to do about bad language on the internet. In *Proceedings of the 2013 annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, 2013.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 1277–1287, 2010.

Jacob Eisenstein, Noah A. Smith, and Eric P. Xing. Discovering sociolinguistic associations with structured sparsity. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics*, ACL, pages 1365–1374, 2011.

Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-Based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 226–231, 1996.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Michael Wimmer, and Xiaowei Xu. Incremental clustering for mining in a data warehousing environment. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB, pages 323–333, 1998.

Stefan Evert. *The statistics of word cooccurrences: Word pairs and collocations*. PhD thesis, University of Stuttgart, 2005.

Federico Michele Facca and Pier Luca Lanzi. Mining interesting knowledge from weblogs: a survey. *Data and Knowledge Engineering*, 53(3):225–241, 2005.

Robert Fano. *Transmission of information: A statistical theory of communications*. The MIT Press, 1961.

Cao Feng, Ester Martin, Qian Weining, and Zhou Aoying. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 6th SIAM International Conference on Data Mining*, SDM, pages 328–339, 2006.

Huiji Gao, Jiliang Tang, and Huan Liu. Exploring social-historical ties on Location-Based Social Networks. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, ICWSM, 2012.

Daniel Gayo-Avello. A meta-analysis of state-of-the-art electoral prediction from Twitter data. *CoRR*, abs/1206.5851, 2012.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for Twitter: annotation, features, and experiments. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics*, ACL, pages 42–47, 2011.

Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2008.

Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for Twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, WWW, pages 593–596, 2013.

Sharad Goel, Jake M. Hofman, Sébastien Lahaie, David M. Pennock, and Duncan J. Watts. Predicting consumer behavior with web search. *Proceedings of the National Academy of Sciences*, 107(41):17486–17490, 2010.

Andre Gohr, Alexander Hinneburg, Rene Schult, and Myra Spiliopoulou. Topic evolution in a stream of documents. In *Proceedings of the 9th SIAM International Conference on Data Mining*, SDM, pages 859–870, 2009.

Scott A. Golder and Michael W. Macy. Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 333(6051):1878–1881, 2011.

Mehmet Gönen and Ethem Alpaydin. Multiple Kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, 2004.

Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):515–528, 2003.

Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

David L.W. Hall, Daniel Jurafsky, and Christopher D. Manning. Studying the history of ideas using topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, 2008.

Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: makn sens a #twitter. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics*, ACL, pages 368–378, 2011.

Bo Han, Paul Cook, and Timothy Baldwin. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL, pages 421–432, 2012.

Bo Han, Paul Cook, and Timothy Baldwin. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology*, 4(1):5:1–5:27, 2013.

Zellig Harris. Distributional structure. *Word*, 10:146–162, 1954.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.

Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H. Chi. Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI, pages 237–246, 2011.

Bill Heil and Mikolaj Piskorski. New twitter research: Men follow men and nobody tweets. *Harvard Business Review*, 1, 2009.

Mark Herbster and Manfred K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 782–792, 2011.

Matthew D. Hoffman, David M. Blei, and Francis R. Bach. Online learning for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems*, NIPS, pages 856–864, 2010.

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, UAI, pages 289–296, 1999.

Courtenay Honeycutt and Susan C. Herring. Beyond microblogging: conversation and collaboration via Twitter. In *Proceedings of the 2009 42rd Hawaii International Conference on System Sciences*, HICSS, 2009.

Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in Twitter. In *Proceedings of the 1st Workshop on Social Media Analytics*, SOMA, 2010.

Mark Edward Huberty. Multi-cycle forecasting of congressional elections with social media. In *Proceedings of the 2nd Workshop on Politics, Elections and Data*, PLEAD, pages 23–30, 2013.

Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. Online multi-scale dynamic topic models. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, 2010.

Bernard J. Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. Twitter power: tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 60(11):2169–2188, 2009.

Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, WebKDD/SNA-KDD, pages 56–65, 2007.

Steven Johnson. How Twitter will change the way we live. *Time Magazine*, 173: 23–32, 2009.

Andreas Jungherr, Pascal Jürgens, and Harald Schoen. Why the Pirate Party won the German Election of 2009 or the trouble with predictions: A response to Tumasjan, A., Sprenger, T. O., Sander, P. G., Welpe, I. M. 'Predicting elections with Twitter: What 140 characters reveal about political sentiment'. *Social Science Computer Review*, 30(2):229–234, 2012.

Ben King and Steven Abney. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the 2013 annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, 2013.

Jon Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 91–101, 2002.

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, pages 591–600, 2010.

Su Mon Kywe, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu. On recommending hashtags in Twitter networks. In *Proceedings of the 4th International Conference on Social Informatics*, SocInfo, pages 337–350, 2012.

Vasileios Lampos. On voting intentions inference from Twitter content: a case study on UK 2010 General Election. *arXiv:1204.0423*, 2012.

Vasileios Lampos and Nello Cristianini. Tracking the flu pandemic by monitoring the social web. In *2nd IAPR Workshop on Cognitive Information Processing*, CIP, pages 411–416, 2010.

Vasileios Lampos, Tijl De Bie, and Nello Cristianini. Flu Detector: tracking epidemics on Twitter. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD '10, pages 599–602, 2010.

Vasileios Lampos, Daniel Preoţiuc-Pietro, and Trevor Cohn. A user-centric model of voting intention from Social Media. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics*, ACL, 2013.

Vasileios Lampos, Nikolaos Aletras, Daniel Preoţiuc-Pietro, and Trevor Cohn. Predicting and characterising user impact on Twitter. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, EACL, 2014.

David Laniado and Peter Mika. Making sense of Twitter. In *Proceedings of the 9th International Semantic Web Conference*, ISWC, pages 470–485, 2010.

Thomas Lansdall-Welfare, Vasileios Lampos, and Nello Cristianini. Effects of the recession on public mood in the UK. In *Proceedings of the 21st International Conference on World Wide Web Companion*, WWW, pages 1221–1226, 2012.

Olivier Ledoit and Michael Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621, 2003.

Chung-Hong Lee. Mining spatio-temporal information on microblogging streams using a density-based online clustering method. *Expert Systems with Applications: An International Journal* , 39(10):9623–9641, 2012.

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, pages 155–164, 2012.

Wei Li and Andrew McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML, pages 577–584, 2006.

Wei Li, Xuerui Wang, and Andrew McCallum. A continuous-time model of topic co-occurrence trends. In *Proceedings of the 2006 AAAI Workshop on Event Extraction and Synthesis*, pages 48–53, 2006.

Defu Lian and Xing Xie. Learning location naming from user check-in histories. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS, 2011.

David Liben-Nowell, Jasmine Novak, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences*, 102(33):11623–11628, 2005.

Frank Lin and William Cohen. A very fast method for clustering big text datasets. In *Proceedings of the 19th European Conference on Artificial Intelligence*, ECAI, pages 303–308, 2010.

Jimmy Lin, Rion Snow, and William Morgan. Smoothing techniques for adaptive online language models: topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 422–429, 2011.

Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. Paraphrasing 4 microblog normalization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 73–84, 2013.

Marco Lui and Timothy Baldwin. Cross-domain feature selection for language identification. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, IJCNLP, pages 553–561, 2011.

Marco Lui and Timothy Baldwin. langid.py: an off-the-shelf language identification tool. In *Proceedings of the 50th annual meeting of the Association for Computational Linguistics*, ACL, pages 25–30, 2012.

Zongyang Ma, Aixin Sun, and Gao Cong. Will this #hashtag be popular tomorrow? In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, 2012.

Zongyang Ma, Aixin Sun, and Gao Cong. On predicting the popularity of newly emerging hashtags in Twitter. *Journal of the American Society for Information Science and Technology*, 64(7):1399–1410, 2013.

James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

Jalal Mahmud, Jeffrey Nichols, and Clemens Drews. Where is this tweet from? Inferring home locations of Twitter users. In *Proceedings of the dixth International Conference on Weblogs and Social Media*, ICWSM, 2012.

Kevin Makice. *Twitter API - up and running: learn how to build Twitter applications.* O'Reilly, 2009.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

Micol Marchetti-Bowick and Nathanael Chambers. Learning for microblogs with distant supervision: political forecasting with Twitter. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL, pages 603–612, 2012.

André F. T. Martins. *The geometry of constrained structured prediction: Applications to inference and learning of Natural Language Syntax.* PhD thesis, Carnegie Mellon University, 2012.

André F. T. Martins, Noah A. Smith, Mario A. T. Figueiredo, and Pedro M. Q. Aguiar. Structured sparsity in structured prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, 2011a.

André F. T. Martins, Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mario A. T. Figueiredo. Online learning of structured predictors with multiple kernels.

In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, AISTATS, 2011b.

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, pages 1155–1158, 2010.

Allie Mazzia and James Juett. Suggesting hashtags on Twitter. 2011. URL `http://www-personal.umich.edu/~amazzia/pubs/545-final.pdf`.

Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman and Hall/CRC, 1989.

Paul McFedries. *Twitter Tips, Tricks, and Tweets*. John Wiley & Sons, 2010.

James McInerney, Alex Rogers, and Nicholas R Jennings. Learning periodic human behaviour models from sparse data for crowdsourcing aid delivery in developing countries. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, UAI, 2013.

Cha Meeyoung, Haddadi Hamed, Benevenuto Fabricio, and P. Gummadi Krishna. Measuring user influence in Twitter: the million follower fallacy. In *Proceedings of the 4th International Conference on Weblogs and Social Media*, ICWSM, pages 10–17, 2010.

Panagiotis Takis Metaxas, Eni Mustafaraj, and Daniel Gayo-Avello. How (not) to predict elections. In *SocialCom/PASSAT*, pages 165–171, 2011.

Greg Miller. Social scientists wade into the tweet stream. *Science*, 333(6051):1814–1815, 2011.

David Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, UAI, 2008.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*, ACL-IJCNLP, pages 1003–1011, 2009.

Jean Jacques Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes Rendus de l'Académie des Sciences (Paris), Série A*, 255: 2897–2899, 1962.

Dhiraj Murthy. Twitter: Microphone for the masses? *Media, Culture & Society*, 33 (5):779–789, 2011.

Eni Mustafaraj, Samantha Finn, Carolyn Whitlock, and Panagiotis Takis Metaxas. Vocal minority versus silent majority: discovering the opionions of the long tail. In , SocialCom/PASSAT, pages 103–110, 2011.

Mor Naaman, Jeffrey Boase, and Chih-Hui Lai. Is it really about me?: message content in social awareness streams. In *Proceedings of the 2010 ACM conference on Computer Supported Cooperative Work*, CSCW, pages 189–192, 2010.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. SemEval-2013 task 2: sentiment analysis in Twitter. In *2nd Joint Conference on Lexical and Computational Semantics (*SEM): Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, *SEM, pages 312–320, 2013.

Radford M. Neal. *Bayesian learning for Neural Networks*. Springer-Verlag New York, Inc., 1996.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Proceedings of the 2010 annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 100–108, 2010.

Andrew Ng, Michael. Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, NIPS, 2002.

Dong Nguyen and Seza Dogruoz. Word level language identification in online multilingual communication. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, 2013.

Thanh Nguyen and Phuong Tu Minh. In *IEEE International Conference on Research, Innovation and Vision for the Future*, 2007.

Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. An empirical study of geographic user activity patterns in Foursquare. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, ICWSM, 2011.

Liadan O'Callaghan, Adam Meyerson, Rajeev Motwani, Nina Mishra, and Sudipto Guha. Streaming-Data Algorithms for High-Quality Clustering. In *Proceedings of the 18th International Conference on Data Engineering*, ICDE, pages 685–694, 2002.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: linking text sentiment to public opinion time series. In *Proceedings of the 4th International Conference on Weblogs and Social Media*, ICWSM, 2010.

Tim O'Reilly and Sarah Milstein. *The Twitter Book*. O'Reilly, 2009.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 380–390, 2013.

Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to Twitter. In *Proceedings of the 2010 annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 181–189, 2010.

Saša Petrović, Miles Osborne, and Victor Lavrenko. RT to win! predicting message propagation in Twitter. In *Proceedings of the 5th International Conference on Weblogs and Social Media*, ICWSM, 2011.

Saša Petrović, Miles Osborne, and Victor Lavreko. I wish I didn't say that! analyzing and predicting deleted messages in Twitter. *CoRR*, abs/1305.3107, 2013.

Hamed Pirsiavash, Deva Ramanan, and Charless Fowlkes. Bilinear classifiers for visual recognition. In *Advances in Neural Information Processing Systems*, volume 22 of *NIPS*, pages 1482–1490, 2009.

Tamara Polajnar, Simon Rogers, and Mark Girolami. Protein interaction detection in sentences via Gaussian Processes: a preliminary evaluation. *International Journal Data Mining and Bioinformatics*, 5(1):52–72, 2011.

Daniel Preoţiuc-Pietro and Trevor Cohn. A temporal model of text periodicities using Gaussian Processes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, 2013a.

Daniel Preoţiuc-Pietro and Trevor Cohn. Mining user behaviours: A study of check-in patterns in Location Based Social Networks. In *Proceedings of the ACM Web Science Conference*, WebSci, 2013b.

Daniel Preoţiuc-Pietro, Sina Samangooei, Trevor Cohn, Nick Gibbins, and Mahesan Niranjan. Trendminer: an architecture for real time analysis of social media text. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media, Workshop on Real-Time Analysis and Mining of Social Streams*, ICWSM, 2012.

Daniel Preoţiuc-Pietro, Justin Cranshaw, and Tae Yano. Exploring venue-based city-to-city similarity measures. In *Proceedings of the 19th ACM SIGKDD 2013, the 2nd International Workshop on Urban Computing (UrbComp 2013)*, SIGKDD, 2013a.

Daniel Preoţiuc-Pietro, Sina Samangooei, Vasileios Lampos, Trevor Cohn, Nicholas Gibbins, and Mahesan Niranjan. Clustering models for discovery of regional and demographic variation. *Public deliverable, Trendminer Project*, 2013b.

Daniel Preoţiuc-Pietro, Sina Samangooei, Andrea Varga, Douwe Gelling, Trevor Cohn, and Mahesan Niranjan. Tools for mining non-stationary data - v2. Clustering models for discovery of regional and demographic variation - v2. *Public deliverable, Trendminer Project*, 2014.

Ignacio Quesada and Ignacio Grossmann. A global optimization algorithm for linear fractional and bilinear programs. *Journal of Global Optimization*, 6(1):39–76, 1995.

Daniel Ramage, Susan Dumais, and Dan Liebling. Characterizing microblogs with topic models. In *Proceedings of the 4th International Conference on Weblogs and Social Media*, ICWSM, pages 130–137, 2010.

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's Razor. In *Advances in Neural Information Processing Systems*, NIPS, 2000.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

Kejiang Ren, Shaowu Zhang, and Hongfei Lin. Where are you settling down: Geolocating twitter users based on tweets and social networks. In *Information Retrieval Technology*, Lecture Notes in Computer Science, pages 150–161, 2012.

Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the Levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking*, 19(3):630–643, 2011.

Miguel Rios and Jimmy Lin. Distilling Massive Amounts of Data into Simple Visualizations: Twitter Case Studies. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media, Workshop on Social Media Visualization*, ICWSM, 2012.

Miguel Rios and Jimmy Lin. Visualizing the 'Pulse' of world cities on Twitter. In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*, ICWSM, 2013.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in Tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, 2011.

Chris Rogers and David Williams. *Diffusions, Markov processes, and martingales. Vol. 1.* Cambridge Mathematical Library. Cambridge University Press, 2000.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI, pages 487–494, 2004.

Dominic Rout, Daniel Preoţiuc-Pietro, Bontcheva Kalina, and Trevor Cohn. Where's @wally: A classification approach to geolocating users based on their social ties. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT, 2013.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing, 2002.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW, pages 851–860, 2010.

Salvatore Scellato and Cecilia Mascolo. Measuring user activity on an online Location-Based Social Network. In *Proceedings of 3rd International Workshop on Network Science for Communication Networks*, NetSciCom, 2011.

Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T. Campbell. NextPlace: A spatio-temporal prediction framework for pervasive systems. In *Proceedings of the 9th International Conference on Pervasive Computing*, Pervasive, 2011a.

Salvatore Scellato, Anastasios Noulas, Renaud Lambiotte, and Cecilia Mascolo. Socio-spatial properties of online Location-Based Social Network. In *Proceedings of 5th International AAAI Conference on Weblogs and Social Media*, ICWSM '11, 2011b.

Juliane Schäfer and Korbinian Strimmer. A hrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology, The Berkeley Electronic Press*, 4 (1), 2005.

Harald Schoen, Daniel Gayo-Avello, Takis Panagiotis Metaxas, Eni Mustafaraj, Markus Strohmaier, and Peter A. Gloor. The power of prediction with social media. *Internet Research*, 23(5):528–543, 2013.

Kashif Shah, Trevor Cohn, and Lucia Specia. An investigation on the effectiveness of features for translation quality estimation. In *MT Summit*, 2013.

Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

David W. Sims, Emily J. Southall, Nicolas E. Humphries, Graeme C. Hays, Corey J. A. Bradshaw, Jonathan W. Pitchford, Alex James, Mohammed Z. Ahmed, Andrew S. Brierley, Mark A. Hindell, David Morritt, Michael K. Musyl, David Righton, Emily L. C. Shepard, Victoria J. Wearmouth, Rory P. Wilson, Matthew J. Witt, and Julian D. Metcalfe. Scaling laws of marine predator search ehaviour. *Nature*, 451(7182):1098–1102, 2008.

Padhraic Smyth and Scott White. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 5th SIAM International Conference on Data Mining*, SDM, pages 76–84, 2005.

Tristan Snowsill, Florent Nicart, Marco Stefani, Tijl De Bie, and Nello Cristianini. Finding surprising patterns in textual data streams. In *Proceedings of the 2nd International Workshop on Cognitive Information Processing*, CIP, pages 405–410, 2010.

Libo Song, David Kotz, Ravi Jain, and Xiaoning He. Evaluating location predictors with extensive Wi-Fi mobility data. *SIGMOBILE Mobile Computing and Communications Review*, 7(4):64–65, 2003.

Xiaodan Song, Ching-Yung Lin, Belle L. Tseng, and Ming-Ting Sun. Modeling and predicting personal information dissemination behavior. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD, pages 479–488, 2005.

Daniel Sousa, Luís Sarmento, and Eduarda Mendes Rodrigues. Characterization of the twitter @replies network: are user ties social or topical? In *Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents*, SMUC, pages 63–70, 2010.

Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, Jose Antonio de Macedo, Fabio Porto, and Christelle Vangenot. A Conceptual View on Trajectories. *Data and Knowledge Engineering*, 65(1):126–146, 2008.

Lucia Specia, Kashif Shah, Jose G. C. de Souza, and Trevor Cohn. QuEst - A translation quality estimation framework. In *Proceedings of the Association of Computational Linguistics (Demonstations)*, ACL, 2013.

Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP, pages 952–961, 2012.

Carlo Strapparava and Alessandro Valitutti. WordNet-Affect: An affective extension of WordNet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, LREC, pages 1083–1086, 2004.

Bongwon Suh, Hong Lichan, Peter Pirolli, and Ed H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in Twitter network. In *Proceedings of the 2010 IEEE 2nd International Conference on Social Computing*, SocialCom, pages 177–184, 2010.

Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition, Fourth Edition.* Academic Press, 2008.

Rob Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, pages 91–108, 2005.

Robert Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Oren Tsur and Ari Rappoport. What's in a hashtag? content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, WSDM, 2012.

Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, and Isabell M. Welpe. Predicting elections with Twitter: What 140 characters reveal about political sentiment. In *Proceedings of the 4th International Conference on Weblogs and Social Media*, ICWSM, pages 178–185, 2010.

Peter Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, ACL, pages 417–424, 2002.

Peter Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, 2010.

Benjamin Van Durme and Ashwin Lall. Streaming Pointwise Mutual Information. In *Advances in Neural Information Processing Systems*, NIPS, 2009.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17 (4):395–416, 2007.

Chong Wang, David M. Blei, and David Heckerman. Continuous time Dynamic Topic Models. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, UAI, 2008.

Chong Wang, Bo Thiesson, Christopher Meek, and Blei David. Markov Topic Models. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, AISTATS, 2009.

Xuerui Wang and Andrew McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, 2006.

Xing Wei, Jimeng Sun, and Xuerui Wang. Dynamic mixture models for multiple time series. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI, pages 2909–2914, 2007.

Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian Process covariance kernels for pattern discovery and extrapolation. In *Proceedings of the International Conference on Machine Learning*, ICML, 2013.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT-EMNLP, pages 347–354, 2005.

Dekai Wu and Pascale Fung. Improving Chinese tokenization with linguistic filters on statistical lexical acquisition. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, ANLC, pages 180–181, 1994.

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, pages 705–714, 2011.

Kevin Xu, Mark Kliger, and Alfred Hero III. Evolutionary spectral clustering with adaptive forgetting factor. In *2010 IEEE International Conference on Acoustics Speech and Signal Processing*, ICASSP, pages 2174–2177, 2010.

Wei Xu, Alan Ritter, and Ralph Grishman. Gathering and generating paraphrases from Twitter with application to normalization. In *Proceedings of the 6th Workshop on Building and Using Comparable Corpora*, pages 121–128, 2013.

Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM, pages 599–608, 2010.

Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, WSDM, 2011.

Yi Yang and Jacob Eisenstein. A Log-Linear Model for Unsupervised Text Normalization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, 2013.

Dani Yogatama, Michael Heilman, Brendan O'Connor, Chris Dyer, Bryan R. Routledge, and Noah A. Smith. Predicting a scientific community's response to an article. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 594–604, 2011.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, 68(1):49–67, 2006.

Eva Zangerle, Wolfgang Gassler, and Gunther Specht. Recommending #-tags in Twitter. In *Proceedings of the Workshop on Semantic Adaptive Social Web*, UMAP, 2011.

Dejin Zhao and Mary Beth Rosson. How and why people Twitter: the role that micro-blogging plays in informal communication at work. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work*, GROUP, pages 243–252, 2009.

Peng Zhao and Bin Yu. On model selection consistency of LASSO. *Journal of Machine Learning Research*, 7(11):2541–2563, 2006.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing Twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR, pages 338–349, 2011.

Max Zimmermann, Irene Ntoutsi, Zaigham Faraz Siddiqui, Myra Spiliopoulou, and Hans-Peter Kriegel. Discovering global and local bursts in a stream of news. In *Proceedings of the 27th annual ACM Symposium on Applied Computing*, SAC, pages 807–812, 2012.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2): 301–320, 2005.