University of BRISTOL

Detecting Events and Patterns in Large-Scale User Generated Textual Streams with Statistical Learning Methods

Vasileios Lampos

Department of Computer Science
University of Bristol

A thesis submitted to the University of Bristol in accordance with the requirements for the degree of **Doctor of Philosophy**

August 2012

“The lunatic is in the hall,
the lunatics are in my hall,
the paper holds their folded faces to the floor,
and every day the paper boy brings more.

And if the dam breaks open many years too soon,
and if there is no room upon the hill,
and if your head explodes with dark forebodings too,
I’ll see you on the dark side of the moon.”

Pink Floyd ∼ Brain Damage

Declaration

I declare that the work in this thesis was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award.

Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such.

Any views expressed in the thesis are those of the author.

Vasileios Lampos

Signature: ....

Date: ....

Abstract

A vast amount of textual web streams is influenced by events or phenomena emerging in the real world. The social web forms an excellent modern paradigm, where unstructured user generated content is published on a regular basis and in most occasions is freely distributed. The present Ph.D. Thesis deals with the problem of inferring information – or patterns in general – about events emerging in real life based on the contents of this textual stream. We show that it is possible to extract valuable information about social phenomena, such as an epidemic or even rainfall rates, by automatic analysis of the content published in Social Media, and in particular Twitter, using Statistical Machine Learning methods. An important intermediate task regards the formation and identification of features which characterise a target event; we select and use those textual features in several linear, non-linear and hybrid inference approaches achieving a significantly good performance in terms of the applied loss function. By examining further this rich data set, we also propose methods for extracting various types of mood signals revealing how affective norms – at least within the social web's population – evolve during the day and how significant events emerging in the real world are influencing them. Lastly, we present some preliminary findings showing several spatiotemporal characteristics of this textual information as well as the potential of using it to tackle tasks such as the prediction of voting intentions.

Acknowledgements

I leave this Ph.D. ‘significantly’ older. This is a matter of fact. All other matters – abstract or not – surrounding this script as well as the time behind it are going to be products of subjective analysis.

I am grateful to my parents, **Ioannis Lampos** and **Sofia Kousidou**, and my brother **Spyros Lampos**. Every person in this world is bound to its ‘family’ in a healthy and limitless way. Well, ‘limitless’ might sometimes be – by definition – unhealthy, but diving properly into this primitive relationship may need another Ph.D. Thesis and surely no ‘funding’ exists for such an adventure.

Next, and, of course, with a significant importance measured, however, on a relatively different scale, I would like to thank **Prof. Nello Cristianini**, my Ph.D. advisor. Keeping all the good moments in mind, it has been fun working with him; it was as if we were not really working.

I would like to express my gratitude to **Prof. Peter Flach** – my internal Ph.D. progress reviewer – for his quite useful advice and remarks in every single progress review.

I am also indebted to **Dr. Steve Gregory** for his help in several occasions during and right after the completion of my M.Sc. studies.

I am grateful to **Dr. Tijl De Bie** and **Prof. Ricardo Araya** for their contribution in my work. I would also like to thank **Thomas Lansdall-Welfare** and **Elena Hensinger** for our short collaboration.

During my time in the university, I came across many interesting people such as the **porters** of the Merchant Venturers Building. I thank them for being polite and positive every time I was in and out from the building. This might sound strange, but positive energy – for some people who are by default negative – can be extremely important.

I would like to thank **Omar Ali**, **Syed Rahman** and **Stefanos Vatsikas** for being honest and helpful in many occasions. I am also grateful to **Christopher Mims**, the journalist who wrote the first article on our work for MIT’s Technology Review. A special ‘thank you’ to **Dr. Philip J. Naylor** for maintaining our servers and databases; he is the person who once literally saved this project.

This Ph.D. project would not have been possible, if **EPSRC** did not grant me a DTA scholarship (DTA/SB1826) to cover my tuition fees and without **NOKIA Research**'s financial support. On the same ground, I would also like to thank the Computer Science Department of the University of Bristol, and its Head at the time **Prof. Nishan Canagarajah**, for covering parts of my funding in the third year of my Ph.D.; I have to thank my family – again – for covering the rest.

My friends – amongst other more important things – were also part of my Ph.D. studies and I 'have' to thank them. Therefore, **Charis Fanourakis**, **Chloé Vastesaeger**, **Costas Mimis**, **Leonidas Kapsokalivas**, **Nick Mavrogiannakis**, **Takis Tsiatsis** and **Thanos Ntzoumanis** there you go, you got your names in my 'book'.

Finally, I would like to thank all those **Twitter people** I have been interacting with for their indirect help in making me understand several aspects of this platform better and, of course, for their invaluable company during the endless nights I spent in writing up. I wish them 'success' in their life quests, if and only if.

Contents

List of Figures

List of Tables

Abbreviations

Acronym	What (it) **S**tands **F**or
API	**A**pplication **P**rogramming **I**nterface
Bagging	**B**ootstrap **Agg**regat**ing**
Bolasso	**Bo**otstrap **L**east **A**bsolute **S**hrinkage and **S**election **O**perator
CART	**C**lassification **A**nd **R**egression **T**ree
CDC	**C**enters for **D**isease **C**ontrol (and Prevention)
CDF	**C**umulative **D**ensity **F**unction
CI	**C**onfidence **I**nterval
CT	**C**onsensus **T**hreshold
DSC	**D**ominant **S**entiment **C**lass
GP	**G**eneral **P**ractitioner
i.i.d.	**I**ndependent and **I**dentically **D**istributed
ILI	**I**nfluenza-**l**ike **I**llness
IR	**I**nformation **R**etrieval
IS	**I**mpact **S**core
JD	**J**accard **D**istance
LAR	**L**east **A**ngle **R**egression
LASSO	**L**east **A**bsolute **S**hrinkage and **S**election **O**perator
Lowess	**Lo**cally **we**ighted **s**catterplot **s**moothing
MAE	**M**ean **A**bsolute **E**rror
MDS	**M**ulti**d**imensional **S**caling
MFMS	**M**ean **F**requency **M**ood **S**coring
MRE	**M**ean **R**anking **E**rror
MSE	**M**ean **S**quare **E**rror
MSFMS	**M**ean **S**tandardised **F**requency **M**ood **S**coring
MTS	**M**ean **T**hresholded **S**entiment
NA	**N**egative **A**ffectivity
NLP	**N**atural **L**anguage **P**rocessing
OLS	**O**rdinary **L**east **S**quares

PA	**P**ositive **A**ffectivity
PCA	**P**rincipal **C**omponent **A**nalysis
PDF	**P**robability **D**ensity **F**unction
POS	**P**art **O**f **S**peech
RCGP	**R**oyal **C**ollege of **G**eneral **P**ractitioners
RMSE	**R**oot **M**ean **S**quare **E**rror
RSS	**R**esidual **S**um of **S**quares
SE	**S**tandard **E**rror
SnPOS	**S**entiment **n**o **P**art **O**f **S**peech (tagging)
SPOS	**S**entiment **w**ith **P**art **O**f **S**peech (tagging)
SPOSW	**S**entiment **w**ith **P**art **O**f **S**peech (tagging) and **W**ordNet's core terms
SS	**S**imilarity **S**core
TF-IDF	**T**erm **F**requency **I**nverse **D**ocument **F**requency
VSM	**V**ector **S**pace **M**odel
VSR	**V**ector **S**pace **R**epresentation
XML	e**X**tensive **M**arkup **L**anguage

Chapter 1

Introduction

In July 2008, Google reported that the World Wide Web is formed by at least one trillion of unique web pages,[1] approximately one hundred times more than their estimated number in 2005 [62]. What seemed obvious to everybody was proven in numbers; the World Wide Web has become immense.

Web, also known as 'the internet', is definitely one of the most important tools in our everyday life. Numerous of our activities, including education, work, socialising and entertainment are now assisted by or transferred entirely to the 'virtual' online space handled by numerous pieces of hardware and software. Together with internet's evolution, the way users interact with it has been evolving as well. For example, while in the past it would have been considered as an abnormal behaviour to give up privacy for socialising in an online medium, nowadays, this is becoming a mainstream way of conducting life [32].

Social networks, such as Twitter or Facebook, apart from boosting the web usage by individuals,[2] also increased significantly the overall amount of information stored on or linked via the web. This fast expansion of the Social Web – which still is under way – means that large numbers of people can publish their thoughts at no cost. Current estimates put the

[1] Official Google blog, "We knew the web was big", July 2008, http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html.

[2] Nielsen News, "Social Networks/Blogs Now Account for One in Every Four and a Half Minutes Online", June 2010, http://goo.gl/DW0e0.

number of Facebook users at 845 million [44][3] and of Twitter active users at 100 million.[4] The result is a massive stream of digital text that has attracted the attention of marketers [76], politicians [33] and social scientists [71]. By analysing the stream of communications in an unmediated way, without relying on questionnaires or interviews, many scientists are having direct access to people's opinions and observations for the first time. Perhaps equally important is that they have access – although indirectly – to situations on the ground that affect the web users, such as a spreading infectious disease or extreme weather conditions, as long as these are mentioned in the messages being published.

The reader might have noticed that the references in the previous paragraph cite fairly recent work. When we started this project, in September 2008, we were ourselves one of the research groups with the desire to explore this rich amount of information and the aforementioned pieces of work were also on the development phase. Our first and very preliminary findings, presented in "Weather talk" [89],[5] were pointing to the conclusion that even web resources, such as blogs, which are not frequently updated and usually do not reflect real time events, enclosed information that under the right modelling could be used to make inferences about moments in real life; in this particular case, we were making predictions about the weather state of several UK locations based on blog posts and news articles.

One could argue that web information might be unreliable, because it is not always "officially verified", and badly structured due to the great variety of ways for organising and publishing content and thus, the task of automatic interpretation is – by definition – hard. By considering that we are also dealing with large-scale amounts of information, this interpretation becomes a non human-feasible process.

During this project, we are forming ways for understanding and explaining the textual content that lies on the web. Understanding covers the notion of identifying key properties in the text, which might in turn express domains of knowledge emerging in the real world. Explaining defines the set of functions, which are in the position to map those key properties to proper inferences that relate to events occurring in those domains. Encapsulated in this

[3] As of 31/12/2011.

[4] Twitter Blog, "One hundred million voices", 08/09/2011. Link: http://blog.twitter.com/2011/09/one-hundred-million-voices.html.

[5] "Weather talk" is the title of the author's M.Sc. Thesis, written right before the beginning of his Ph.D. project. That work provided interesting insights, but the developed methods described or the data used in the M.Sc. project were not considered in the project presented by this Thesis.

process is the discovery of patterns, which simplify and merge those large amounts of data into simpler and representable conclusions.

1.1 Summary of questions, aims and goals

This research project is applied on the overlapping scientific fields of Artificial Intelligence, Machine Learning, Data Mining, Pattern Recognition and Information Retrieval. Those fields offer a plethora of well developed techniques; we are using the state-of-the-start, but also try to achieve adaptations or improvements in methods, algorithms or frameworks relevant to the context of our work, wherever is needed. In this section, we made an effort to break up the main purpose of this project – already expressed in the previous section – into sub-questions or tasks ending up with a more specific and constrained list of aims and goals.

Does useful information exist in the 'open' web text streams? While this question might seem simple to address, every answer must be followed by proper justification. A simplistic hypothesis would, of course, favour the case that in such large amounts of textual information, the probability of not encountering subsets with a decreased uncertainty and hence, of high informational value should be very small. Indeed, several works – including ours – proved that statement in practice during the early stages of this Ph.D. project. Those findings reshaped the initial question to the following one.

Supposing that such information does exist, how can it be successfully extracted from the text stream? The answer to this question is essentially the main point of interest for this Ph.D. project as a substantial part of it has been dedicated in developing methods for deriving conclusions or conducting inferences by processing text. Methodologies are formed by combining or extending models from the fields of Machine Learning, Information Retrieval (**IR**) and Natural Language Processing (**NLP**). The fact that we will be handling massive amounts of data, makes almost certain the application of techniques that aim to reduce the dimensionality of the problem. 'Useful' information might consider specific events or more general patterns arising in the population.

Is this extracted information a good representation of one – at least – fraction of real life? Measuring the performance of a method is a mainstream task in statistical learning performed by defining one or more loss functions. In this case though, it is necessary to numerically quantify this fraction of real life; it might consider an event that can be measured in a variety of units or an even more abstract notion, such as a public reaction, emotion or opinion on real-time happenings. In a supervised learning framework, the existence of ground truth is essential not only for learning the parameters of a model, but also for testing the inferred learning function. On the other hand, other more abstract scenarios, enclosed in pattern discovery tasks, usually seek confirmation by either proving their statistical significance or by querying 'common sense'.

Formulate generic methodologies. In this project we aim to discover generalised methodologies that are not solving only one specific problem, but are applicable to a class of problems. This could be achieved through an experimental process which investigates case studies with different characteristics. A general method should be applicable to a class of problems and thus, such a class must be carefully defined.

Improve various subtasks. Depending on the success in developing methods able to address the aforementioned questions and aims, we should also consider how the latter can be improved. Our methods may be multilayered since they will not only consider text processing, but also learning and inference schemes. The extraction and selection of textual features or the algorithms for learning and inference might play an important role in this process and hence, improvements should always be a part of our research.

1.2 Work dissemination

Several parts of the work presented in this Thesis have attracted the interest of mainstream, technology or health news media. In this section, we refer to some significant articles that disseminated our research to the general public as they provide the reader with well-shaped introductory information about it as well as with a perspective different from the academic one.

- "How Twitter could better predict disease outbreaks", in MIT Technology Review by Christopher Mims (14/07/2010). Link: http://www.technologyreview.com/blog/mimssbits/25480/ – Article featuring our paper [93].

- “Engines of the future: The cyber crystal ball”, in New Scientist by Phil McKenna (27/11/2010). Link: http://goo.gl/roBa2.

- “Could Social Media Be Used to Detect Disease Outbreaks?”, in Science Daily. Press release from the University of Bristol (1/11/2011). Link: http://www.sciencedaily.com/releases/2011/11/111101125812.htm – Article featuring our paper [91].

- “2011: the year in review”, EPSRC’s annual review (23/12/2011). Link: http://www.epsrc.ac.uk/newsevents/news/2011/Pages/yearinreview.aspx#may.

- “Brits study forecasting flu outbreaks using Twitter”, in Examiner by Linda Chalmer Zemel (27/12/2011). Link: http://goo.gl/7auob – Article featuring our paper [91].

- “Diagnosing flu symptoms with social media”, in Natural Hazards Observer (Vol. XXXVI(4), pp. 7–9) by Vasileios Lampos and Nello Cristianini (March 2012) [90]. Link: http://www.colorado.edu/hazards/o/archives/2012/mar12_observerweb.pdf.

- “Can Twitter Tell Us What We Are Feeling?”, in Mashable by Sonia Paul (17/04/2012). Link: http://mashable.com/2012/04/17/twitter-feeling/ – Article featuring our paper [95].

Our work has also had some impact on current research as several recent publications from various research groups build on or cite it; we refer to some of them in the last chapter of the Thesis.

1.3 Peer-reviewed publications

In this section, we list all relevant peer-reviewed publications in chronological order co-authored by Vasileios Lampos during his Ph.D. project. We have defined three levels of contribution and tagged all listed publications according to them:
— Written the paper (**W**)
— Contributed in ideas, methods, or algorithms (**M**)
— Provided data (**D**)

Tracking the flu pandemic by monitoring the Social Web. **Vasileios Lampos** and Nello Cristianini. IAPR Cognitive Information Processing (CIP 2010), Elba Island, Italy. IEEE

Press, 2010. — **W, M, D**
In this paper, we presented the first – though preliminary – method for using Twitter content to infer flu rates in a population (see Chapter 4).

Flu Detector – Tracking Epidemics on Twitter. **Vasileios Lampos**, Tijl De Bie and Nello Cristianini. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010), Barcelona, Spain. Springer, 2010. — **W, M, D**
This paper presented Flu Detector (http://geopatterns.enm.bris.ac.uk/epidemics/), a tool for detecting flu rates in several regions in the UK by exploiting Twitter content. The method described in this paper is an improvement of the one presented in the previous publication (see Chapter 8).

Nowcasting Events from the Social Web with Statistical Learning. **Vasileios Lampos** and Nello Cristianini. ACM Transactions on Intelligent Systems and Technology (TIST), Vol. 3(4), 2011. — **W, M, D**
In this paper, we present a general methodology for nowcasting the occurrence and magnitude of an event or phenomenon by exploring the rich amount of unstructured textual information on the Social Web. The paper supports its findings with two case studies: a) inference of flu rates for several UK regions and b) inference of rainfall rates for several locations in the UK (see Chapter 5).

Effects of the Recession on Public Mood in the UK. Thomas Lansdall-Welfare, **Vasileios Lampos**, and Nello Cristianini. Mining Social Network Dynamics (MSND) session on Social Media Applications in News and Entertainment (SMANE) at WWW '12, Lyon, France in February 2012. — **M, D**
In this paper, we associate mood scores with real-life events in the UK. Amongst other interesting findings, we show that there exists evidence which connects recession in the UK with increased levels of negative mood in the Social Media (see some relevant findings in Section 6.2).

Detecting Temporal Mood Patterns in the UK by Analysis of Twitter Content. **Vasileios Lampos**, Thomas Lansdall-Welfare, Ricardo Araya, and Nello Cristianini. Under submission process to the British Journal of Psychiatry (April 2012). — **W, M, D**
In this paper, we analyse seasonal circadian affective mood patterns based on Twitter content (see Section 6.1).

1.4 Summary of the Thesis

After this introductory section, in **Chapter 2** we define the theoretical background on which our work has been based on. This includes basic notions from Machine Learning and Information Retrieval; some even more fundamental notions – usually considered as common knowledge – have been included in **Appendix A**; **Appendix B** contains tables with input data used in several experimental processes of ours. In **Chapter 3**, we refer to the data collection methods used to crawl the social network of Twitter, as well as provide the reader with a characterisation of this Social Network. **Chapter 4** describes the first steps of our work, where we developed a supervised learning method able to infer flu rates by using the content of Twitter as input information [92]. **Chapter 5** builds on the previously presented methodology, trying to resolve its limitations and also prove its generalised capacity by applying it to another task: the inference of rainfall rates from the content of Social Media [91]. In **Chapter 6**, we are aiming to extract affective mood patterns from the Social Web. We focus on two tasks: the extraction of circadian and daily mood patterns and we look into ways able to interpret them [94, 95]. **Chapter 7** presents some preliminary and unpublished work regarding three Pattern Discovery tasks on Twitter content; the first one investigates content-based relations among locations and proposes a method for forming geo-temporal content similarity networks, the second analyses users' posting time patterns and the third one presents methodologies which aim to model and infer voting intention polls [88]. **Chapter 8** presents two tools we developed that showcased our work to the general public and the academic community: Flu Detector [93] and Mood of the Nation. We conclude this Thesis in **Chapter 9**, referring also to possible future work.

Chapter 2

Theoretical Background

In this chapter we go through some fundamental theoretical notions which serve as a basis for our work. We start by defining the general field of this research project, that is Machine Learning. Regression and Classification, the most common supervised learning tasks, are discussed in short; we draw our interest into regularised regression as well as classification and regression trees. Bootstrap and Bagging, two methods for enhancing the learning process, are also presented and discussed. We move on by presenting some basic notions and methods from the field of Information Retrieval, such as the Vector Space Model and stemming. Finally, we define feature extraction and selection, two important tasks that have been performed quite often in our work.

2.1 Defining the general field of research

"*Learning is remembering*", says Socratis[1] but Platonas[1] makes the inverse statement, "*remembering is learning*." Aristotelis[1] expands the notion of learning by claiming that "*each form of teaching or mental learning begins from the knowledge that already exists*."

Learning, therefore, could be conceived as a process of acquiring new knowledge or other characteristics, such as a specific skill, by fusing input information with experience and previous knowledge.[2] The Greek philosophers were probably making statements primarily referring to humans or animals, but nowadays the notion of learning could also refer to a machine or a computer program.

Several definitions have been given to **Machine Learning** in the past years. Arthur Samuel – broadly known for his pioneering work on developing computer programs able to play the game of checkers [142, 143] – defined it as a branch of Artificial Intelligence focused on the task of making computers learn without being explicitly programmed. Tom Mitchell in [115] pointed out that Machine Learning is a discipline that addresses the following general question: "*How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?*" In his previous work [114] he had already formalised the main concept of Machine Learning with the following statement: "*A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.*" Consequently, in contrast to animals, the level of learning in machines can be quantified in a non abstract manner; actually, improving based on a performance metric serves as the main proof of learning.

In several Machine Learning tasks one tries to identify and exploit patterns in data using algorithms primarily driven by the statistical approach. The roots of Machine Learning come from the general field of Computer Science, whereas some of its most popular theoretical tools derive from Statistics, and therefore it is related or even quite often also referred to as **Statistical Learning** [10, 67]. Over the past years, three basic categories of learning have been defined: supervised, unsupervised and reinforcement learning.

[1] Also – commonly – known as Socrates, Pluto and Aristotle respectively.

[2] A modified version of the definition given in Wikipedia. Link: http://en.wikipedia.org/wiki/Learning.

In a supervised learning scenario we are provided with a set of inputs $\mathcal{X}$ (also known as observations or evidence), a corresponding set $\mathcal{Y}$ of targets or responses and the main task is to learn a function f which maps the inputs to the targets, $f : \mathcal{X} \to \mathcal{Y}$. In unsupervised learning there exist no specific targets; the main task there could be generally described as trying to discover a pattern enclosed in the input data by exploiting their content. Finally, reinforcement learning aims to learn from interaction, for instance by discovering which actions yield higher rewards after trying them [156].

In this work, we apply or derive methodologies which mainly belong to the supervised learning class. We also apply basic methods from unsupervised learning or other generic statistical algorithms in our effort to extract patterns from data. In the next sections, we give a short description of the theory and methods that have been used as a basis throughout this project.

2.2 Regression

Regression is a supervised learning task, where the unknown variable y, also known as the target variable, is real valued, that is $y \in \mathbb{R}$. In some cases, the target variable can be n-dimensional, $y \in \mathbb{R}^n$ for some $n \in \mathbb{N}$, though with loss of important information, we can break up this task to n one-dimensional independent regression scenarios.

2.2.1 Linear Least Squares

Suppose that knowing the mapping function $f : \mathcal{X} \to \mathcal{Y}$, we are given an input set in the form of a vector $X = [x_1, x_2, ..., x_n]$, $X \in \mathcal{X}$ and we want to predict a real-valued output $y \in \mathcal{Y}$. The linear regression model takes the following form:

$$y = f(X, w_0, w) = w_0 + \sum_{i=1}^{n} x_i w_i, \tag{2.1}$$

where f is defined by n coefficients or weights (w_1, ...,w_n) one for each dimension of $\mathcal{X}$ and one global intercept or bias, denoted by w_0. The target variable (also known as the response) y is a linear combination of the observations and the elements of the mapping function.

Suppose now that we have a set of m n-dimensional observations held in a $m \times n$ matrix X, where x_{ij} is the j-th element of the i-th row, and their corresponding m targets as contents of a vector y. The most common way to learn the mapping function f, known as Ordinary Least Squares (OLS) regression, is by trying to minimise the Residual Sum of Squares (RSS):

$$\text{RSS}(w_0, w) = \sum_{i=1}^{m} \left(y_i - w_0 - \sum_{j=1}^{n} x_{ij} w_j \right)^2 . \tag{2.2}$$

Therefore, the solution is given by

$$[w_0^*, w^*] = \arg\min_{w_0, w} \text{RSS}(w_0, w). \tag{2.3}$$

In the equivalent vector notation and by adding a column of 1's to the left side of X (thus the first element of w is the intercept), we have that:

$$\text{RSS}(w) = (y - Xw)^T (y - Xw). \tag{2.4}$$

By differentiating RSS with respect to w and setting the derivative equal to zero we can derive a closed-form solution for the ordinary least squares:

$$[w_0^*, w^*] = w = (X^T X)^{-1} X^T y. \tag{2.5}$$

This closed-form solution reveals an obvious limitation of the OLS regression: $X^T X$ may be singular and thus difficult to invert, resulting in inconsistent solutions [68]. Further analysis and interpretation of OLS indicates that least squares estimates often have low bias but large variance which affects negatively their prediction accuracy (see Appendix A.2 for the definitions of a predictor's variance and bias). Additionally, when the input is high-dimensional, OLS generates a large number of predictors (often as much as the dimensions), something that complicates interpretation [67].

2.2.2 Regularised Regression

Regularisation is a common practice to encounter the limitations of OLS regression. It can be seen as adding a penalty term to the original error function (Equation 2.2) of OLS regression:

$$\begin{aligned} \mathrm{E}(w_0, w) &= \mathrm{RSS}(w_0, w) + \lambda \|w\|_q && (2.6) \\ &= \sum_{i=1}^{m} \left(y_i - w_0 - \sum_{j=1}^{n} x_{ij} w_j \right)^2 + \lambda \sum_{j=1}^{n} |w_j|_q, && (2.7) \end{aligned}$$

where λ (≥ 0) is known as the regularisation parameter and q decides which norm of w is being used. Parameter λ can be interpreted as a coefficient that controls the relative importance between the data dependent error (RSS) and the regularised one. The amount of regularisation is increased as λ increases. In literature, regularisation is also referred to as shrinkage since by minimising $\mathrm{E}(w_0, w)$ one also aims to 'shrink' the regression coefficients w_j. It is not common to regularise the bias term w_0 as this would remove its main purpose which is to provide a prior knowledge for the target variable.

In a high-dimensional linear regression model many variables might be correlated. Consequently, a large positive coefficient on a variable x_j could be cancelled by a large negative coefficient on $x_{j'}$ given that x_j and $x_{j'}$ are correlated. By shrinking regression's coefficients and therefore imposing a size constraint on them, one can restrict this effect [67].

By setting $q = 0$ we are performing a variable **subset selection** – that is finding a subset of predictors which gives the smallest RSS – since penalty's sum will count the number of nonzero coefficients. This approach provides sparse solutions, but is not computationally feasible for problems of high dimensionality. Still, there exist algorithms, which can handle a small number of predictors [15, 51] or hybrid approaches, such as Forward or Backward Stepwise Selection [35]. The common disadvantage of all those learning models is that being discrete processes, they exhibit high variance and quite often do not reduce the prediction error of the full OLS regression model.

For $q = 2$, Equation 2.6 defines the minimisation target of **ridge regression**, also referred to as L2-norm regularisation given that the L2-norm of w is minimised [68]. Similarly to OLS, ridge regression has a closed form solution given by:

$$[w^*, w_0^*] = (X^T X + \lambda C)^{-1} X^T y, \tag{2.8}$$

where a column of 1's has been added to the left side of X to accommodate w_0^* and C is a an identity matrix with a 0 as its top diagonal element. Another way to look at ridge regression is to observe that a positive constant λ has been added to the diagonal of $X^T X$ to deal with singularity problems of the original matrix [14], a process also known as Tikhonov regularisation [161]. However, regularising the L2-norm does not encourage sparsity, *i.e.* regression's coefficients are nonzero, and therefore ridge regression does not offer an easily interpretable model.

2.2.3 Least Absolute Shrinkage and Selection Operator (LASSO)

On the other hand, for $q = 1$, *i.e.* by regularising the L1-norm, the advantages of regularisation are maintained and at the same time sparse solutions are retrieved. This regularisation is known as *Least Absolute Shrinkage and Selection Operator* or the **LASSO** [158]. The original formulation of LASSO is the following optimisation problem:

$$[w_0^*, w^*] = \arg\min_w \sum_{i=1}^{m} \left(y_i - w_0 - \sum_{j=1}^{n} x_{ij} w_j \right)^2 \text{ subject to } \sum_{j=1}^{n} |w_j| \leq t, \tag{2.9}$$

where t controls the amount of regularisation (or shrinking) of the weight vector's L1-norm. This can be easily adapted to the previous formulation,

$$[w_0^*, w^*] = \arg\min_w \left\{ \sum_{i=1}^{m} \left(y_i - w_0 - \sum_{j=1}^{n} x_{ij} w_j \right)^2 + \lambda \sum_{j=1}^{n} |w_j| \right\}. \tag{2.10}$$

As in the previous cases, λ controls the amount of regularisation. Setting $\lambda = 0$ transforms LASSO to OLS, whereas using a very large λ will completely shrink w to zero leading to a null model. A moderate choice of λ will result to the cancellation of some variables of the initial model by setting their corresponding regression coefficients to zero.

In contrast to ridge regression, LASSO does not have a closed form solution. Computing the solution of LASSO is a quadratic programming problem and estimations of the solution can be derived within this framework [8]. A more efficient way to compute LASSO is achieved by enforcing a modification on the Least Angle Regression (LAR) algorithm [39].

The modified LAR (see algorithm 1) computes the entire regularisation path of LASSO, *i.e.* all LASSO solutions for different choices of the regularisation parameter λ. In a nutshell,

ALGORITHM 1: Least Angle Regression (LAR) modification for LASSO

1. Standardise X so that each predictor (or column) $X_{:,j}$ has a zero mean and variance equal to 1. Start with the residual set to $r = y - \bar{y}$ and all regression coefficients to zero, w_1, w_2, ..., $w_p = 0$.
2. Find $X_{:,c}$ the most correlated predictor with r.
3. Move w_c from zero towards its least-squares coefficient $\langle X_{:,c}, r \rangle$ until some other predictor $X_{:,i}$ has as much correlation with the current (active) residual r_a as $X_{:,c}$.
4. Move (w_c, w_i) in the direction defined by their joint least squares coefficient of the current residual on $(X_{:,c}, X_{:,i})$, until some other predictor $X_{:,\ell}$ has as much correlation with the current residual. If a nonzero coefficient becomes zero, then drop its corresponding variable and recompute the current joint least squares direction.
5. Iterate through this process until all p predictors have been entered. If the predictors are more or equal to the observations ($p \geq N$), then LAR completes after $N - 1$ steps.

LAR firstly sets all regression coefficients to zero and searches for the predictor (or observation) $X_{:,c}$ that has the highest correlation with the residual $r = y - \bar{y}$, where $\bar{y}$ denotes the mean of y. Then explores the direction of this predictor until an other predictor $X_{:,i}$ has as much correlation with the current residual. It proceeds in a direction equiangular between the two predictors until a third variable $X_{:,\ell}$ enters the active set of most correlated predictors. Then again it continues along the least angle direction of $X_{:,c}$, $X_{:,i}$ and $X_{:,\ell}$ until a fourth variable enters the most correlated set and so on. LASSO modification proposes dropping variables when their corresponding coefficients arrive to zero from a nonzero previous value during this process [67].

LASSO has a unique solution assuming that there exists no pair of predictors which are perfectly collinear [159]. Recent work has also shown that LASSO has a unique solution (with probability one), if the predictor variables are drawn from a continuous probability distribution [160]. LASSO is shown to be sign-consistent – meaning that sign as well as zeros in the inferred w are consistent –, if there are low correlations between relevant and irrelevant variables [168, 171, 173, 174]. However, it is not model-consistent as in many settings (for example when predictors are strongly correlated) it does not recover the true model, *i.e.* errors are made in the assignment of nonzero weights to variables and therefore, the inferred sparsity pattern differs from the true one [103, 173]. There exist various proposed ways to fix the model inconsistency of LASSO, such as the adaptive [174] or relaxed LASSO [106], thresholding [100], bootstrapping [4] and others [107, 169].

2.3 Classification

Classification is a supervised learning task where a model, known as the *classifier*, is constructed to predict categorical labels. Recall that the prediction target in regression is a numeric value. In classification, we aim to infer the label of a class, for example "Category A" or "Category B" and so on. Those categories can be represented by discrete numerical values, *i.e.* 1 and 2 for Categories A and B respectively [65]. Using the notation of section 2.2, the target variable y is a set of m labels ℓ, $y = \{\ell_1, ..., \ell_m\}$, $m \in \mathbb{N}$, and by mapping labels to different integers, it can also take a numeric representation, $y \in \mathbb{Z}$. We are aiming to learn a function $f : \mathcal{X} \rightarrow y$, where $\mathcal{X}$ denotes the set of observations. In this work, we do not directly solve classification problems; however, we are using structures – such as the Classification and Regression Trees (see Section 2.3.2) – that were primarily built to solve such problems, but later on were extended to include regression problems as well.

2.3.1 Decision Trees

A decision tree is a structure that classifies instances of observable data by sorting them down the tree using a top-down approach, starting from the root and finishing at a leaf node. Leaf nodes are labels indicating the result of the classification for this instance. A node of the decision tree targets a specific attribute of the instance and depending on the value of this attribute, it indicates which descending branch the decision making process should follow [114].

Figure 2.1 shows an example of a very basic decision tree. Supposing that we have a function able to compute a flu-score from the Social Media content on the web – the higher the flu-score, the bigger the risk for an epidemic –, and at the same time no other official information from health authorities exists about the actual flu rates in the population, the purpose of this decision tree is to figure out whether a warning for a spreading flu epidemic should be issued or not. Of course, this is a very simplified example to help the reader understand how decision trees function.

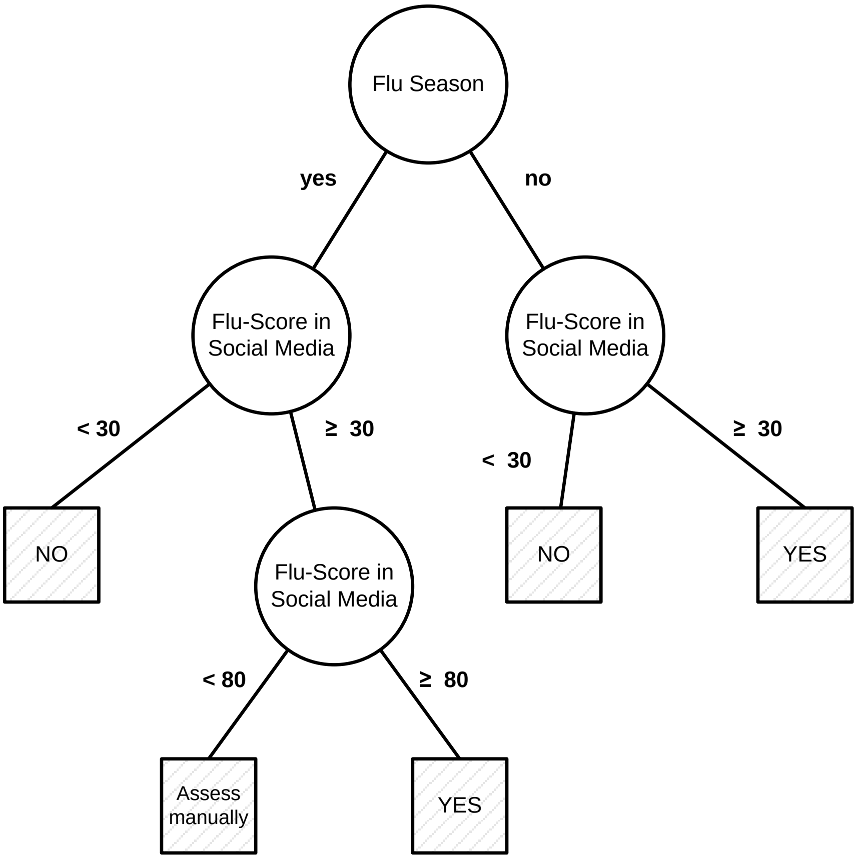

Flu Season
yes
no
Flu-Score in Social Media
Flu-Score in Social Media
< 30
≥ 30
< 30
≥ 30
NO
Flu-Score in Social Media
NO
YES
< 80
≥ 80
Assess manually
YES

Figure 2.1: A simplified decision tree for answering the following question: "Based on the Social Media content, should we issue a warning for a possible flu epidemic?"

Suppose that we are within a typical 'flu season' and from our data we compute a flu score equal to 90. The instance

$$\langle\textbf{Flu Season}\text{: yes, }\textbf{Flu-Score in Social Media} = 90\rangle$$

would be sorted down the middle leaf of the decision tree and would therefore be classified as 'YES', meaning that a warning should be issued. Likewise, the instance

$$\langle\textbf{Flu Season}\text{: yes, }\textbf{Flu-Score in Social Media} = 75\rangle$$

would be classified as 'Assess manually', which in practice means that human supervision is needed in order to make a decision.

Decision trees have found application in text mining [1], computational linguistics [146], failure diagnosis [24], the medical domain [81, 150] and so on. Many scientific fields such as Statistics, Artificial Intelligence (Machine Learning) or Decision Theory have developed methodologies for learning/constructing automatically decisions trees from data [118]. Over the past years, several approaches (and the respective tools) have been developed for decision tree learning, such as **CHAID** (CHi-squared Automatic Interaction Detector) [80], the well-known **ID3** [133] and its descendant **C4.5** [132], **QUEST** (Quick Unbiased Efficient Statistical Tree) [99] and many more. In the next section, we will focus briefly on one approach, the Classification and Regression Tree (**CART**) [16] as we have chosen to use this one in our work (see Chapter 5).

The main advantage that decision trees have is their interpretability. When their size is reasonable, they do create self-explanatory models that can be easily understood by non-experts. Furthermore, decision tree algorithms have been developed able to handle discrete, continuous variables as well as missing values and perform classification or regression. In addition, decision trees are a non-parametric method, which means that they do not rely on assumptions made about the probability distribution that creates the data [137]. However, the construction of optimal decision trees is an NP-complete problem [73] and therefore the approximate result of learning might not always be optimal. Apart from the non-optimality, decision trees also suffer from instability; minor changes in the training set, might result to the construction of a very different model [16, 132].

2.3.2 Classification and Regression Tree

In this section, we explain briefly the main operations of Classification and Regression Tree (CART), which is a sophisticated program for constructing trees from data developed by Breiman *et al.* and presented in [16]. In addition to the original script, this section was also based on review papers about CART [97, 155].

CART is a non-parametric method capable of solving classification and regression problems, which not only has had a significant coverage in the literature (of several academic fields), but also a great number of applications (see for example [47, 96, 131, 136] and many more). Since it is a well established method, there also exist off-the-shelf software packages implementing variations of it, such as the one in MATLAB [105].

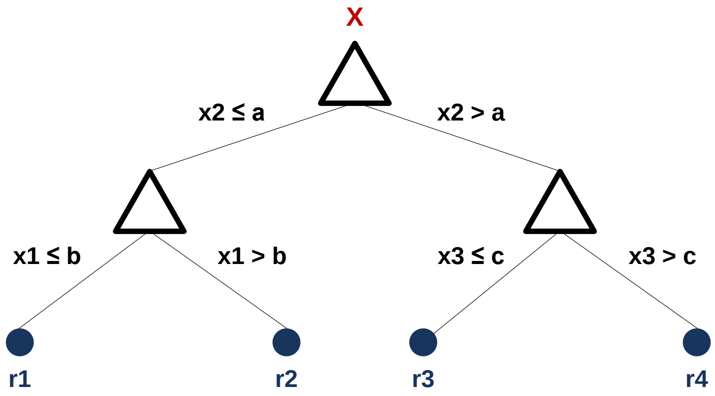


Figure 2.2: An example of a classification and regression tree (CART)

Figure 2.2 shows a very simple example of a CART with four terminal nodes. $\mathcal{X}$ denotes the entire space of input variables; specific variables (such as x_1, x_2 or x_3) are used in the decision making process similarly to the previous section's decision tree example. Given that the terminal nodes of the tree are real-valued numbers, this tree is used to perform regression; a similar tree could perform classification if the terminal nodes were class labels.

CART is constructed by splitting $\mathcal{X}$ into two subsets and then by recursively splitting $\mathcal{X}$'s subsets and so on. By forming two splits of each subset, CART constructs a binary tree. This should not be considered as a limitation due to the fact that binary trees can be used to represent any other type of tree structures. During the tree construction process, when for example $\mathcal{X}' \subset \mathcal{X}$ is divided into two subsets, there exists no rule forcing to divide those two new subsets using the same variable. This is made more clear by observing Figure 2.2, where the left subset is divided by x_1 and the right one by x_3. There is also no limitation in using the same variable for creating subsets in different branches of the tree as well in the same branch for more than one time.

The core question about the construction of a CART is how to best split the subsets of $\mathcal{X}$. Supposing that y is the set of response values for the training set $\mathcal{X}$, each split of a subset $\mathcal{X}' \subset \mathcal{X}$ to a left ($\mathcal{L}$) and a right ($\mathcal{R}$) sub-subset is decided by solving the following optimisation problem:

$$\{\mathcal{L},\mathcal{R}\} = \underset{\mathcal{L},\mathcal{R}}{\arg\min} \left(\sum_{i:\, x_i \in \mathcal{L}} (y_i - \bar{y}_{\mathcal{L}})^2 + \sum_{i:\, x_i \in \mathcal{R}} (y_i - \bar{y}_{\mathcal{R}})^2 \right), \tag{2.11}$$

where $\bar{y}_{\mathcal{L}}$ and $\bar{y}_{\mathcal{R}}$ are the sample means of the response values in y corresponding to $\mathcal{L}$ and $\mathcal{R}$ respectively. The value of a terminal node t of the regression tree is set as the mean response value $y_{\mathcal{X}_t}$ that corresponds to the subset $\mathcal{X}_t \subset \mathcal{X}$.

Deciding the optimal size of the tree is controlled by an error-complexity metric $C_\alpha(T)$ defined as:

$$C_\alpha(\mathcal{T}) = C(\mathcal{T}) + \alpha|\mathcal{T}|, \tag{2.12}$$

where $|\mathcal{T}|$ denotes the size of the regression tree, $C(\mathcal{T})$ is sum of squared errors (or of any other loss function applied) and α is a parameter that controls the ratio between the size of the tree and its loss on the learning sample. Similarly to LASSO, $\alpha|\mathcal{T}|$ can be seen as a regularisation parameter that penalises overfitting. By using an independent validation set and a loss function – usually Mean Square Error (**MSE**) –, one can further prune the tree and decide its optimal and final version.

2.4 Bootstrap

Bootstrap was introduced as a method for assessing the accuracy of a prediction but has also found applications in improving the prediction itself [37, 38, 40]. Suppose that we have a sample of n observations $X = [x_1, x_2, ..., x_n]$, all of the same probability $1/n$. Based on X, as well as on a predictor function f, we are able to calculate an estimate $\hat{y} = f(X)$ of the actual response y. The primary aim of bootstrap was to estimate how accurate the prediction $\hat{y}$ is.

A bootstrap sample $B = [x_1^*, x_2^*, ..., x_n^*]$ is a random sample of size n drawn with replacement from X. x_i^*'s notation denotes that it is different from the original x_i; the bootstrap data set B consists of members of X, some appearing zero times, some appearing once, some appearing more than once and so on. Algorithm 2 shows how Bootstrap is used to estimate the Standard Error (**SE**) of a predictor f, one of its primary applications.

2.4.1 Bagging

Bootstrap **Aggregating**, known as **Bagging**, is a method for generating multiple versions of a predictor and then using them to get an aggregated and possibly better – in terms of

ALGORITHM 2: Bootstrap algorithm for estimating standard error

1. Form M independent bootstrap samples B_m each consisting of n elements drawn with replacement from X.
2. Evaluate the bootstrap replication corresponding to each bootstrap sample
$$\hat{y}_m = f(B_m),\, m = 1, 2,\ \ldots,\ M. \tag{2.13}$$
3. Estimate the SE($\hat{y}$) by computing the standard deviation of the M replications
$$\text{SE}(\hat{y}) = \sqrt{\left(\sum_{m=1}^{M} (\hat{y}_m - \text{E}(\hat{y}))^2\right) / (M-1)}. \tag{2.14}$$

performance – predictor. Those versions are the result of carrying out learning on bootstrap replicates of the original learning set. When predicting a numerical value, Bagging averages over the versions of the predictor, whereas when the prediction target is categorical, it uses a plurality vote to decide the outcome of the prediction [17]. The main operations of Bagging are also also described in Algorithm 3.

Bagging improves the MSE for unstable predictors, but does not have a significant impact on stable schemes; a predictor is unstable, if small changes in the data can cause large changes in the predicted values [18]. Bagging might also improve MSE for sufficiently large sample sizes, increases the squared bias [21] and reduces variance for higher order terms leaving the linear term unaffected [50]. Those two properties of reducing MSE and variance have an important effect on hard decision problems, *e.g.* an estimation of a regression function after testing in regression and classification trees; bagging is shown to smooth such hard decisions [19]. Interestingly, its properties do not necessarily depend on the dimensionality of the problem [19, 20].

2.5 Feature Extraction and Selection

Feature extraction aims to create new features based on transformations or combinations of an original set of raw attributes. Given a feature space Ω of dimensionality d, feature extraction methods determine an appropriate subspace Ω' of dimensionality $m \leq d$, in a

ALGORITHM 3: Bootstrap Aggregating (Bagging) for estimating a better predictor function

```
1. Suppose we are given a learning set $\mathcal{L} = \{X, y\}$, where $X$ and $y$ denote observations and responses respectively.
2. Take $M$ bootstrap samples from $\mathcal{L}$, $\mathcal{L}_m = \{X_m, y_m\}$, $m = 1, 2, \ldots, M$.
3. Learn $M$ predictor functions $f_m$ using $\mathcal{L}_m$.
4. Compute $M$ inference sets $y_m^{(t)}$ using $f_m$ on a set of unseen observations $X^{(t)}$, $y_m^{(t)} = f_m(X^{(t)})$.
5. (a) If $y$ is a categorical variable, then use a voting scheme to decide the dominant class among $y_m^{(t)}$.
   (b) If $y$ is a numerical variable, then $y^{(t)} = \mathrm{E}(y_m^{(t)})$.
```

linear or non-linear manner [75]. Feature extraction is also a special form of dimensionality reduction, and finds use in many applications in the areas of bioinformatics, text processing, speech processing and so on [64].

Feature selection, on the other hand, is a dimensionality reduction method which aims to select a subset of features or dimensions in an effort to overcome high dimensionality and overfitting, improve the interpretability of the inferred model, and in some learning scenarios speed up the learning and inference process. Feature selection may follow feature extraction; feature extraction is used to define an original set of dimensions or attributes and then feature selection specifies which ones are the most important.

A very basic form of feature selection can be achieved via correlation analysis. Consider one of the feature candidates as a vector x, and the corresponding target as a vector y, both of size n. One of the most primitive feature selection techniques is to measure the **correlation coefficient** $\rho(x, y)$ which is defined as the covariance of (x, y) divided by the product of their standard deviations:

$$\rho(x,y) = \frac{\sum_{i=1}^{n}(x_i - x_\mu)(y_i - y_\mu)}{\sqrt{\sum_{i=1}^{n}(x_i - x_\mu)^2 \sum_{i=1}^{n}(y_i - y_\mu)^2}}, \tag{2.15}$$

where x_μ and y_μ are the first moments of x and y respectively. The lower and upper bounds of ρ are -1 and 1, indicating perfect anti-correlation or correlation (in a linear manner) respectively, whereas a $\rho = 0$ indicates the absence of linear correlation. A correlation coefficient can be used to rank candidate features, but only separately; it is a univariate method.

An important part of our work included the optimal selection of textual features for representing a target concept. Recent research has shown that the bootstrapped version of LASSO, conventionally named as *Bolasso*, by intersecting the supports of bootstrap estimates addresses LASSO's model selection inconsistency problems [4]. We have applied a modification of Bolasso (see Section 5.4.4) in our effort to select a consistent subset of textual features. In addition, a bagged version of CART (also referred to as an ensemble of CARTs) has been applied as an alternative nonlinear and nonparametric method, which deals with the instability problems of CART and at the same time is able to – indirectly – select a subset of important features from a text stream and then use them to solve a regression problem (see Section 5.9).

2.6 Foundations of Information Retrieval

Information Retrieval (**IR**) is a scientific field with its main goal set to provide methods for retrieving useful information from large collections of unstructured material (usually text) [104]. In our context, IR is an automated process based on well defined algorithms and is performed by computer programs. In the following sections, we go through some basic notions of IR used and applied throughout our work.

2.6.1 Vector Space Model

A corpus is a collection of documents, which in turn are collections of words or n-grams (where n denotes the number of words). To convert all this information into something more structured that can receive queries, an algebraic representation is needed. A Vector Space Model (**VSM**) defines such a representation.

For an m-sized set of documents $\mathcal{D} = \{d_1, ..., d_m\}$, which are formed by an n-sized set of n-grams or a vocabulary $\mathcal{V} = \{v_1, ..., v_n\}$, the weight of an n-gram v_i in a document d_j is denoted as w_{ij}. Therefore a document d_j can be represented as a vector or a set of weights:

$$d_j = \{w_{1j}, ..., w_{nj}\}. \tag{2.16}$$

There exist several types of VSMs; the simplest one is the Boolean, which sets a weight equal to 1 if an n-gram exists in a document, otherwise sets it to 0:

$$w_{ij} = \begin{cases} 1 & \text{if } v_i \in d_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.17}$$

Alternatively, one can count the occurrences of the n-gram in the document and normalise this count with the total number of n-grams in the document

$$w_{ij} = \frac{\text{times } v_i \text{ appears in } d_j}{\text{number of } n\text{-grams in } d_j}. \tag{2.18}$$

This weighting scheme is commonly known as Term Frequency (**TF**) VSM.

The VSMs presented above treat each document as a set of n-grams; they do not exploit possible relationships or semantic connections between the n-grams, but consider them as separate and independent random variables. For this reason they are usually referred to as "bag of words" models.

2.6.2 Term Frequency – Inverse Document Frequency

Term Frequency Inverse Document Frequency (**TF-IDF**) is a VSM for document representation which enables to approximate the actual importance of a word or phrase in a document or several documents [140]. Similarly to the previously presented VSMs, TF-IDF is also a "bag of words" VSM, but instead of simply counting the terms in a document and assigning frequencies to them, it calculates a normalised frequency over a set of documents in order to define weights of importance.

For the following description we have used [98, 141] as additional references. Suppose we have a vocabulary $\mathcal{V}$ with n terms and a set $\mathcal{D}$ with m documents, $\mathcal{V} = \{v_1, ..., v_n\}$ and $\mathcal{D} = \{d_1, ..., d_m\}$. For $\forall v_i \in \mathcal{V}$, we calculate the number of its appearances in $\forall d_j \in \mathcal{D}$. As a result, we derive a set $\mathcal{F}$ of size $n \times m$ which contains the raw frequencies or counts of the vocabulary terms in the documents, $\mathcal{F} = \{f_{11}, ..., f_{n1}, f_{12}, ..., f_{ij}, ..., f_{nm}\}$.

The frequency tf_{ij} of a term v_i in a document d_j is given by

$$\text{tf}_{ij} = \frac{f_{ij}}{\max\{f_{1j}, ..., f_{nj}\}}. \tag{2.19}$$

It is a normalised frequency which depends on the maximum raw frequency of a term in a specific document. For a term v_i and a document d_j, if the raw term frequency is equal to zero, $f_{ij} = 0$ (meaning that the term does not appear in the document), then for the respective normalised frequency holds also that $\text{tf}_{ij} = 0$. At this stage, we have retrieved the set $\mathcal{TF}$ which includes all the normalised term frequencies for the documents, $\mathcal{TF} = \{\text{tf}_{11}, ..., \text{tf}_{n1}, \text{tf}_{12}, ..., \text{tf}_{ij}, ..., \text{tf}_{nm}\}$.

$\forall v_i \in \mathcal{V}$ we calculate the total number of documents in which it appears at least once. This is denoted by the document frequency df_i. All the document frequencies are represented by the set $\mathcal{DF} = \{\text{df}_i, ..., \text{df}_n\}$. In order to retrieve the inverse document frequencies set $\mathcal{IDF}$, we use the following formula

$$\text{idf}_i = \log\left(\frac{m}{\text{df}_i}\right), \tag{2.20}$$

where m is the number of documents. Similarly, the $\mathcal{IDF}$ set has n elements, $\mathcal{IDF} = \{\text{idf}_i, ..., \text{idf}_n\}$.

In the final stage of the scheme, we build a set W with weights for every term-document pair, $\mathcal{W} = \{w_{11}, ..., w_{n1}, w_{12}, ..., w_{ij}, ..., w_{nm}\}$. Each w_{ij} is the result of

$$w_{ij} = \text{tf}_{ij} \times \text{idf}_i. \tag{2.21}$$

2.6.3 Text Preprocessing – Stemming and Stop Words

It is a quite common principle to preprocess raw text before forming a vector space model. A well-known method for text preprocessing is known as **stemming** and it is a process that reduces a word to its stem, that is the basis of the word. In that way, many variants of the same word are merged in one variable, something that helps in reducing significantly the size of the vocabulary index and therefore the dimensionality of a text mining problem.

Several approaches to stemming have been proposed [49] with the most popular of them using suffix stripping [101, 129]. Suffix stripping could be defined as the removal or homogeneous calibration of a word’s suffix. In our work, we are applying one of those approaches, the Porter Stemmer, which has been developed in 1980 and later on has been made available through a stemming-specific language (Snowball) as a software package [130]. Some examples of stemming using the Porter algorithm are listed below:

RESEARCH	⟶	RESEARCH	HAPPY	⟶	HAPPI
RESEARCHES	⟶	RESEARCH	HAPPIER	⟶	HAPPIER
RESEARCHED	⟶	RESEARCH	HAPPIEST	⟶	HAPPIEST
RESEARCHER	⟶	RESEARCH	HAPPINESS	⟶	HAPPI
RESEARCHERS	⟶	RESEARCH			
RESEARCHING	⟶	RESEARCH			

In the first group of examples, the stemmer removes any suffix and keeps the core stem ‘RESEARCH’. In the second series, it converts the last character of ‘HAPPY’ from ‘Y’ to ‘I’ and therefore creates the same stem for ‘HAPPY’ and ‘HAPPINESS’, but keeps the words ‘HAPPIER’ and ‘HAPPIEST’ as they are, *i.e.* the stemmer makes a distinction between the comparative and superlative forms of an adjective. On the other hand, since stemming is an automated method based on a set of rules, there might be occasions where it generates significant information loss. For example, both ‘SINGULAR’ and ‘SINGULARITY’ are stemmed to ‘SINGULAR’ or both ‘AUTHOR’ and ‘AUTHORITY’ are stemmed to ‘AUTHOR’.

In addition to stemming, it is desirable to remove words that do not have any particular semantic notion mostly because they are used quite often. Those words are commonly referred to as **stop words** and some examples are articles, *e.g.* ‘a’, ‘an’, ‘the’, propositions, *e.g.* ‘about’, ‘after’, ‘on’, and so on. For English text, there exist several off-the-shelf lists with stop words with the one proposed by Fox in 1989 being the most popular; this particular list was generated by combining the most frequent English words with a set of manual rules encapsulated in a deterministic finite automaton [48]. In most IR tasks, stop words are automatically identified by setting a threshold in the maximum allowed frequency of a word – similarly, spam stop words are decided by setting a threshold on the minimum frequency of a word [104]. In our work, depending on the task at hand, we either use an off-the-shelf stop word list, or set thresholds for the minimum and maximum frequency of a ‘legitimate’ word or use a combination of both practices. The main motivation behind removing stop words is again the reduction in the dimensionality of the problem, but also the existing experimental proof for increased performance [149].

2.7 Summary of the chapter

This chapter gave a summary of the theoretical background behind our research; Appendix A covers some more basic notions. The core scientific field of our work is defined by Artificial Intelligence and more specifically by one of its branches, Machine Learning. Supervised learning, which formulates the scenario where we are given a set of input data X and their corresponding responses y and we have to figure out a function f mapping X to y, gathers the main focus in our work, as we apply regularised regression methods throughout it. LASSO is a regularised regressor providing sparse solutions, however it is shown to be inconsistent. CARTs formulate a method able to provide sparse regression solutions which are nonlinear and nonparametric, but are also unstable. Bolasso, the bootstrap version of LASSO, resolves some model inconsistencies of LASSO; likewise, Bagging addresses the instabilities of CARTs. In our project, we have also incorporated some basic tools from the field of Information Retrieval. Feature extraction approaches create attributes from a raw set of unstructured information, whereas vector space models define ways for giving those attributes a consistent algebraic form. When processing textual information, keeping a stem instead of the entire word in addition to removing stop words helps in reducing the initial dimensionality of a problem; feature selection methods are statistical approaches that can further reduce the original dimensions during learning.

Chapter 3

The Data – Characterisation, Collection and Processing

The mission of this chapter is two-fold. First, we give a short description on the Social Network platform which has been used as the input data space for this research project and also explain its value as an information source. Then, we refer to the methods and tools that have been implemented or incorporated in our data collection and processing routines. Of course, this project is not about collecting data; however, our research would have been impossible without it.

3.1 Twitter: A new pool of information

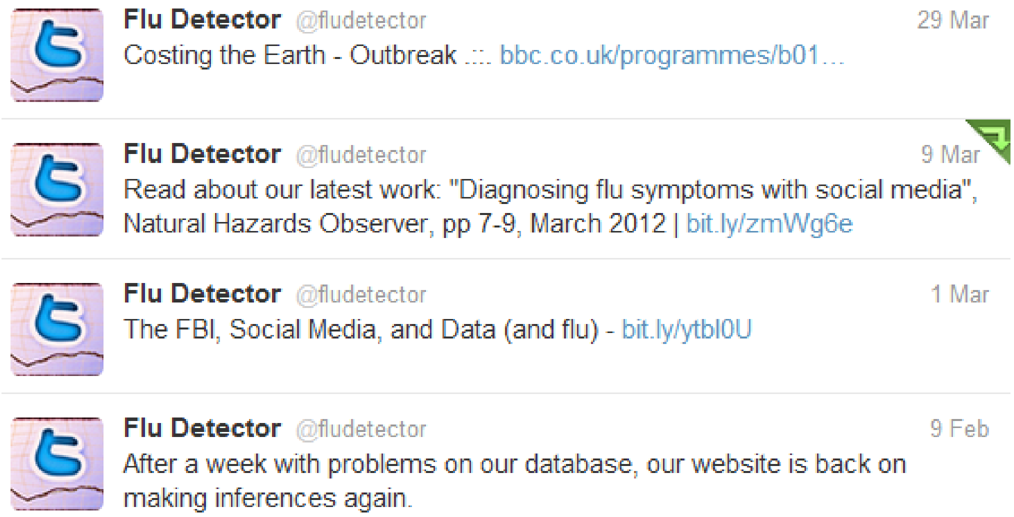
A **blog**, a term derived from synthesising the words web and log, is a – or occasionally a part of a – website supposed to be updated with new content from time to time.[1] A blog's content is based on posts identified primarily by their unique Uniform Resource Locators (URLs). Those posts can vary from random or opinionated articles to the embedding of songs, videos or other multimedia types.

Microblogging is a compact form of blogging, which can be seen as a 'technological evolution' of blogs, and surely is one the best complementary tools for the blogosphere. It became very popular when web services like Tumblr[2] and Twitter[3] kicked off. The basic characteristic (or limitation) of microblogs is that authors are restricted to a specific number of characters per post. For the experimental derivations of the work presented in this Thesis, we collect and use Twitter content.

Twitter was created in March 2006 and by 2011 the number of its users reached 100 million worldwide with a reported new accounts per day ratio equal to 460,000 for February 2011.[4] The registered users of Twitter are allowed to post messages, commonly known as tweets (see Figure 3.1), which can reach a maximum of 140 characters. By default an account is public not private, *i.e.* everybody can see the tweets published by this account without any authorisation by its owner. Users have also the option to follow other users and therefore see their tweets on their time-line, the main part of Twitter's web interface. For public accounts, becoming a follower does not need any type of approval from the followee; private accounts are able to authorise their followers. In that sense, each user – with a public or a private account – has a set of people who he follows (followees) and a set of people who follow him (followers). The only distinction is that connections in public accounts can be seen as one-sided relationships, whereas connected private accounts form a 2-sided – but hidden – relationship.

Users can mention or reply to each other in their tweets – by using the character '@' followed by the username of the user to be mentioned – and therefore, conduct electronic conversations. They can also reproduce the tweet of another user, an action known as retweet.

[1] Definition taken from Wikipedia on 31.10.2011 - http://en.wikipedia.org/wiki/Blog.
[2] Tumblr founded in 2007, http://tumblr.com.
[3] Twitter launched in 2006, http://twitter.com.
[4] Twitter Blog, #numbers - http://blog.twitter.com/2011/03/numbers.html, 14.03.2011.



Flu Detector @fludetector 29 Mar
Costing the Earth - Outbreak .::. bbc.co.uk/programmes/b01...

Flu Detector @fludetector 9 Mar
Read about our latest work: "Diagnosing flu symptoms with social media", Natural Hazards Observer, pp 7-9, March 2012 | bit.ly/zmWg6e

Flu Detector @fludetector 1 Mar
The FBI, Social Media, and Data (and flu) - bit.ly/ytbl0U

Flu Detector @fludetector 9 Feb
After a week with problems on our database, our website is back on making inferences again.

Figure 3.1: This is how the published tweets of the Twitter account (or user) @fludetector look like.

Twitter incorporates topics in tweets; any word that starts with a hash (#) is perceived as something that denotes a topic (*e.g.* #OlympicGamesLondon2012).

3.1.1 Characterisation of Twitter – Why Twitter content is important

"*If you think about our society as being a big organism this is just another tool to look inside of it*", says Noah Smith about Twitter [145]. In this section, we refer to several works which justify this statement in practice and highlight the importance of Twitter content.

Twitter's social network is comprised by two underlying sub-networks: a very dense one made up by followers and followees and a sparser one, where more close and probably real friends participate. The latter one is more influential as expected, yet the former reveals the great degree of connectivity in Twitter [71]. In this space of data, three distinct categories of users have been identified: broadcasters, acquaintances (who are the vast majority) and miscreants (stalkers or spammers) [85].

Interestingly, Kwak *et al.* show that Twitter deviates from known human social networks having a non-power law follower distribution, a short effective diameter and low reciprocity[5] [87]. Twitter users (in the United States) are also a highly non-uniform sample

[5] **Reciprocity** is a quantity that characterises directed graphs by determining the degree of mutual connections – double links with opposite directions – in the graph [53].

of the population and represent a highly non-random sample of the overall race/ethnicity distribution [113].

An early work of Java *et al.* has shown that people use microblogging to talk about their daily activities and to seek or share information. An additional derivation presented in this paper was that users with similar intentions are more likely to be connected with each other [77]. As a result, in this environment, conversationality is also promoted. But Twitter is not only a means for public interaction; it finds use in collaborative scenarios as well [69]. By posting tweets or conducting conversations on Twitter a feeling of connectedness is sustained in working environments [172] and several aspects of educational schemes are improved [61]. Twitter encourages free-flowing and just-in-time interaction between and among students and faculty [36], finds great applicability during conferences [135] and is a great tool for promoting second language active learning methods [12].

The majority of topics on Twitter (approx. 85% in 2010) are headline news or persistent news in nature [87] and methods have been developed enabling the detection of breaking news [144]. Twitter also facilitates communication between communities with different political orientations. Similarly to real-life, an extremely limited connectivity between left- and right-leaning users is reported [28].

Furthermore, Twitter is a rich source for opinion mining and sentiment analysis [121]. Consumer opinions are public for companies to track and then adapt their overall branding strategies based on this intelligence source [76]. Polls can be easily conducted regarding consumer confidence and political opinion [27]. TV stations have started to combine their broadcasts with social networks to improve interaction. In particular, Twitter can provide a better understanding of sentiment's temporal dynamics as a reaction to a political debate [33]. Predicting the result of elections based on Twitter content is a much harder problem to solve, still some preliminary approaches have already been proposed on this topic [54, 88, 102, 109, 164].

Due to the fact that microblogging usually expresses a real-time state of its author, tweets are considered more valuable than other media for connecting this information to personal experiences or situations of the users [172]. Exploiting this fact and Twitter's deviating behaviour during mass convergence and emergency events [72], one can build applications to improve situational awareness during those events [166]. Alternatively, one can deploy methodologies that turn tweets to predictions about signals emerging in real life [91]. For example, work is concentrated in detecting the occurrence and magnitude of disease outbreaks in the population [30, 92, 93, 148] or natural disasters such as an earthquake [139].

From the above, one may naturally come to a conclusion about the high degree of importance that Twitter content has. The interesting, one-sided nature of relationships, the retweet mechanism which allows for a rapid information spread, the ability to conduct online conversations and, on top of all that, the open Twitter API making this source of information easy to crawl, offer an opportunity to study human behaviour [87].

3.2 Collecting, storing and processing Twitter data

At the early stages of this work, there was no Twitter data set to work with and try our models. This is the main reason which forced us to create a pipeline that is able to collect and store tweets. With hindsight, this was an essential step.[6] In this short section, we describe the data collection process with references to external libraries or tools that have been used.

3.2.1 RSS and Atom feeds

Really Simple Syndication (**RSS**)[7] or simply a feed is a family of web formats based on Extensive Markup Language (**XML**) used for publishing and distributing web content in a standardised manner. A more recent development, the **Atom** Syndication Format, is an effort to fix the limitations of RSS by supporting, for example, XML namespaces.[8] Twitter uses Atom feeds to deliver its content in a structured format; one has the option to submit a query on Twitter's API and retrieve Atom feeds which contain sets of tweets matching this query. An example of an Atom feed retrieved from Twitter is shown in Figure 3.2. The main content – the actual tweet – of the feed lies inside the tags `<entry>` and `</entry>`. The purpose of the encapsulated sub-tags is self explanatory, for example `<published>` holds the posting date and time for a tweet.

[6] In the recent past, Twitter data sets used in academic studies have become publicly available. However, soon after they have been removed possibly because the open distribution of such content is against the company's policy.

[7] Or originally RDF Site Summary, where RDF stands for Resource Description Framework.

[8] From Wikipedia: "XML namespaces are used for providing uniquely named elements and attributes in an XML document", http://en.wikipedia.org/wiki/XML_namespace.

```
<feed xmlns:google="http://base.google.com/ns/1.0"
xmlns:openSearch="http://a9.com/-/spec/opensearch/1.1/"
xmlns="http://www.w3.org/2005/Atom" xmlns:twitter="http://api.twitter.com/"
xmlns:georss="http://www.georss.org/georss" xml:lang="en-US">
  <id>tag:search.twitter.com,2005:search/</id>
  <link type="text/html" href="http://search.twitter.com/search?q="
  rel="alternate"/>
  ...
  <link type="application/atom+xml" href="http://search.twitter.com/search.atom?
  page=2&max_id=1000009&q=&geocode=51.516667%2C-0.1%2C5km&lang=en&rpp=100"
  rel="next"/>
  <entry>
    <id>tag:search.twitter.com,2005:1000009</id>
    <published>2011-06-06T20:03:05Z</published>
    <link type="text/html" href="http://twitter.com/@username/statuses/status_id"
    rel="alternate"/>
    <title>Hello Twitter!</title>
    <content type="html">Hello Twitter</content>
    ...
    <google:location>London City</google:location>
    <twitter:lang>en</twitter:lang>
    <author>
      <name>@username (First Name/Last Name)</name>
      <uri>http://twitter.com/@username</uri>
    </author>
  </entry>
  <entry>...</entry>
  ...
</feed>
```

Figure 3.2: A sample Atom feed retrieved by querying Twitter API

3.2.2 Collecting and storing tweets

For data collection, a basic crawler has been implemented in Java programming language; Java was preferred to other object-oriented programming frameworks mainly because it is a well-established programming interface for which many useful methods have already been made available by third-parties. Tweets are collected by querying Twitter's Search API.[9] Based on the fact that information geolocation is a key concept in our studies, we are only considering geolocated tweets, *i.e.* tweets for which the author's location is known to Twitter.

In order not to exceed the request limit set by Twitter, something that usually causes suspensions in the data collection process, we limit our data collection to tweets geolocated

[9] Twitter Search API, https://dev.twitter.com/docs/api/1/get/search.

within a 10Km radius of the 54 most populated urban centres in the UK.[10]

Here is an example query on Twitter's Search API:

http://search.twitter.com/search.atom?
geocode=51.5166,-0.1,10km&
lang=en&
rpp=100&
result_type=recent

With this query, we are able to retrieve the most recently published (`result_type=recent`) 100 tweets (`rpp=100`) geolocated in a 10Km radius from a location with latitude $= 51.5166$ and longitude $= -0.1$ (`geocode=51.5166,-0.1,10km` – this is an approximation for the centre of London) written in English language (`lang=en`). Twitter's response will produce an Atom feed that follows the format shown in Figure 3.2. We collect and parse the XML content on this feed by using the Java libraries in ROME[11] and then store its contents – if they have not crawled before – in a MySQL database.[12] Therefore, for each request we make, we can at most retrieve 100 new entries (tweets) to be stored in the database.

From the above, it becomes apparent that a sampling strategy has been followed during data collection. Every n minutes, we form a search query for each one of the 54 locations and retrieve the 100 latest tweets. The value of n varied from 1 to 10 minutes during this project and was mostly dependent on the total number of tweets we aimed to collect on a daily basis. Obviously, the shortest the collection period (n), the more tweets are being collected; as the number of Twitter users reached very high figures, we were able to reduce this frequency and still collect a large amount of tweets per day. Nevertheless n is kept the same for all urban centres and therefore, possible sampling biases of this type are reduced.

465,696,367 tweets (geolocated in the UK) have been collected by the crawler and then stored in the database from the 19th of June 2009 to the 31st of December 2011 (**926 days**).

[10] The cities considered in alphabetical order are the following: *Aberdeen, Basildon, Belfast, Birmingham, Blackburn, Blackpool, Bolton, Bradford, Brighton, Bristol, Bournemouth, Cardiff, Coventry, Derby, Dundee, Edinburgh, Glasgow, Gloucester, Huddersfield, Hull, Ipswich, Leeds, Leicester, Liverpool, London, Luton, Manchester, Middlesbrough, Newcastle, Newport, Northampton, Norwich, Nottingham, Oldham, Oxford, Peterborough, Plymouth, Poole, Portsmouth, Preston, Reading, Rotherham, Sheffield, Slough, Southampton, Southend, Stockport, Stoke, Sunderland, Swansea, Swindon, Watford, Wolverhampton, York.* More information is also available in Appendix B.1.

[11] ROME Atom/RSS Java utilities, `http://java.net/projects/rome/`.

[12] MySQL Relational Database Management System, `http://www.mysql.com/`.

Those tweets have been published from at most **9,706,618 unique users**; the latter number is an approximation as we did not track users that might have continued to post from the same Twitter account but with a changed username.[13] Within those dates, a rough average number of tweets per user is approx. 50; 502,912 tweets have been collected per day on average. However, this is not a very good representation for the daily data collection figures as the number of registered Twitter users has been constantly increasing together with the published volume of content. To provide the reader with a better picture, we have plotted the daily volume of collected tweets in Figure 3.3(a) as well as its cumulative equivalent in Figure 3.3(b). From the figures, we observe that half-way through the considered period the number of collected tweets starts to increase with a higher acceleration and at a point we start collecting more than one million tweets per day. To reduce possible side-effects induced by the increasing volume of tweets and maintain an equal representation of the Twitter corpus between different dates, we tend to introduce a normalisation factor in our vector space models, which is usually equal to the number of tweets considered (see for example the methods in the Chapters 4 and 5).

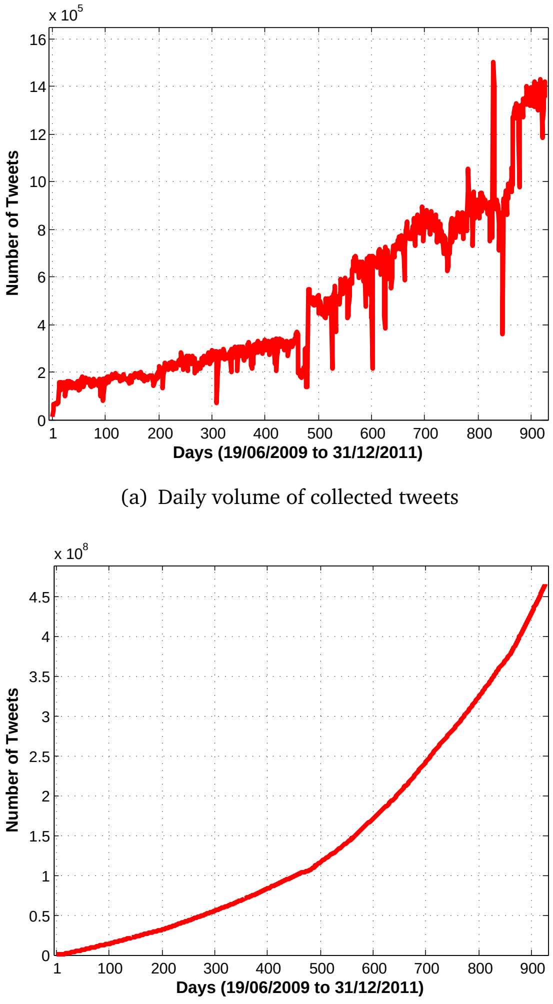
3.2.3 Software libraries for processing textual and numerical data

Several software libraries have been implemented for applying basic methods and algorithms driven by the fields of IR, Data Mining and Machine Learning. A very common software package, used in most projects that deal with the analysis of textual content is Apache Lucene,[14] a set of libraries – initially programmed in Java – able to implement a text search engine. We use Lucene to index tweets, *i.e.* to tokenise and identify unique words in the textual stream, but also for counting word frequencies in documents (a document can be a superset of tweets). Optionally, Lucene can also create indices of stemmed text by applying for example Porter's algorithm [129]. Based on the fact that we already use the Java connector (or driver)[15] to query the database, it is almost straightforward to embed Lucene's libraries in our source code and create several types of indices.

[13] The research in this project did not study the behaviour of individual users, but was focused only on sets of published content.

[14] Apache Lucene, http://lucene.apache.org/.

[15] MySQL Connector/J is the official JDBC driver for MySQL, http://dev.mysql.com/downloads/connector/j/5.0.html.



(a) Daily volume of collected tweets

(b) Cumulative volume plot

Figure 3.3: Volume and cumulative volume plots for tweets collected from the 19th of June, 2009 to the 31st of December, 2011.

Another useful library, which is again a project of Apache Software Foundation, is Mahout.[16] Mahout is capable of handling large scale data sets and implements the most common Machine Learning algorithms for clustering and classification or linear algebra operations such as Singular Value Decomposition. The key about Mahout is that its methods are compatible

[16] Apache Mahout, http://mahout.apache.org/.

with Apache Hadoop[17] in that they are also using the map-reduce paradigm,[18] and therefore, can perform operations using a cluster of computers; when a cluster is not available Mahout can also operate using a single node. We use Mahout to retrieve vector space representations directly from a Lucene index, as there exist already implemented interfaces for this purpose. The VSM can be comprised by term TF or TF-IDF weights as well as other neat options which allow us to remove stop words or rarely used words. In some occasions, Mahout is also used for simple operations such as the computation of cosine similarities between pairs of large vectors.

Given the fact that sometimes we needed customised operations with our data which were not – at least explicitly – implemented in Lucene or Mahout, we have also created our own software libraries (in Java) which can be used to create a text index, perform stemming, remove stop words, compute vector space representations and finally perform our algorithms. Methods from MATLAB®'s native toolboxes as well as from PMTK3[19] have been also used in this work.

3.3 Ground truth

Ground truth is a necessary set of data for our work since many of our proposed methods are based on supervised learning. Hence, ground truth is not only needed for validation, but most importantly for learning the parameters of a model during the training process. We have used three types of ground truth:

- Influenza-like Illness (**ILI**) rates from the Health Protection Agency (**HPA**)[20], the Royal College of General Practitioners (**RCGP**)[21] [46] and the QSurveillance scheme (**QSur**).[22]
- Precipitation (rainfall) measurements from amateur weather stations in the UK. As no official authority could be found to provide us – timely – with rainfall indications,

[17] Apache Hadoop, http://hadoop.apache.org/.
[18] About Map Reduce, http://en.wikipedia.org/wiki/MapReduce.
[19] Probabilistic Modelling Toolkit, http://code.google.com/p/pmtk3/.
[20] Health Protection Agency, http://www.hpa.org.uk/.
[21] Royal College of General Practitioners, http://www.rcgp.org.uk/.
[22] QSurveillance, University of Nottingham and Egton Medical Information Systems Ltd, http://www.qresearch.org/Public/QSurveillance.aspx.

we used information collected by independent weather stations in Bristol, London, Middlesbrough, Reading and Stoke-on-Trent.

- Voting intention polls prior to the 2010 General Election in the UK from YouGov.[23]

The first two sets of data (flu and rainfall rates) are used for the experimental process of Chapters 4 and 5, whereas voting intention polls are used in the preliminary work presented in Section 7.3.

3.4 Summary of the chapter

In this chapter, we started by defining the terms blog and microblog. We moved on by describing the main operations and characteristics of Twitter, a microblogging service which serves as the main source of information for this project. Twitter has attracted the focus of the academic community in the last few years as it allows researchers to study human behaviour using massive amounts of data. This platform has found also use in many other applications such as opinion mining, event detection and education. Next, we described in detail the way we use Twitter's Search API in order to retrieve tweets geolocated in 54 urban centres in the UK; we also provided the reader with figures reporting the average number of collected tweets per day and their cumulative distribution. Then, we referred to some of the most important software tools that we have used to process text, form vector space representations and carry out statistical operations. Finally, we gave a short description of the various versions of ground truth used throughout this project.

[23] YouGov Document Archive, `http://labs.yougov.co.uk/publicopinion/archive/`.

Chapter 4

First Steps on Event Detection in Large-Scale Textual Streams

Tracking the spread of an epidemic disease like seasonal or pandemic influenza is an important task that can reduce its impact and help authorities plan their response. In particular, early detection and geolocation of an outbreak are important aspects of this monitoring activity. Various methods are routinely employed for this monitoring, such as counting the consultation rates of general practitioners. We report on a monitoring methodology capable of measuring the prevalence of disease in a population by analysing the contents of social networking tools, such as Twitter. Our method is based on the analysis of hundreds of thousands of tweets per day, searching for symptom-related statements or automatically identifying potentially illness-related terms, and turning statistical information into a flu-score. We have tested it in the UK for 24 weeks during the H1N1 flu pandemic. We compare our flu-score with data from the Health Protection Agency, obtaining on average a statistically significant linear correlation. This preliminary method uses completely independent data to that commonly used for these purposes, and can be used at close time intervals, hence providing inexpensive and timely information about the state of an epidemic. This chapter is an extended version of our paper "Tracking the flu pandemic by monitoring the Social Web" [92], which – to our knowledge –, is the first one showing that content from the Social Media can be used to detect and infer the rate of an illness in the population.

4.1 Introduction

Monitoring the diffusion of an epidemic in a population is an important and challenging task. Information gathered from the general population can provide valuable insight to health authorities about the location, timing and intensity of an epidemic, or even alert the authorities of the existence of a health threat. Gathering this information, however, is a difficult as well as resource-demanding procedure.

Various methods can be used to estimate the actual number of patients affected by a given illness, from school and workforce absenteeism figures [119], to phone calls and visits to doctors and hospitals [41]. Other methods include randomised telephone polls, or even sensor networks to detect pathogens in the atmosphere or sewage ([74], [110]). All of these methodologies require an investment in infrastructure and have various drawbacks, such as the delay due to information aggregation and processing times (for some of them).

Recent research has pointed out that the use of search engine data could detect geographic clusters with a heightened proportion of health-related queries, particularly in the case of ILI [56]. It demonstrated that timely and reliable information about the diffusion of an illness can be obtained by examining the content of search engine queries.

The work presented in this chapter extends that concept, by monitoring the content of social-web tools such as Twitter, a micro-blogging website, where users have the option of updating their status with their mobile phone device. These updates, commonly known as tweets, are limited to 140 characters only, similarly to the various schemes for mobile text messaging. We analyse the stream of data generated by Twitter in the UK and extract from it a score that quantifies the diffusion of ILI in various regions of the country. The score, generated by applying Machine Learning technology, is compared with official data from the HPA, with which it has a statistically significant linear correlation coefficient that is greater than 95% on average.

The advantage of using Twitter to monitor the diffusion of ILI is that it can reveal the situation on the ground by utilising a stream of data (tweets) created within a few hours, whereas the HPA releases its data with a delay of 1 to 2 weeks. Furthermore, since the source of the data is entirely independent of search engine query-logs or any of the standard approaches, our method can also be used in combination with them, to improve accuracy.

4.2 An important observation

In early stages of this work, an important observation was made. The idea was simple: "What if one used the frequencies of a set of words in Twitter to pick signals emerging in real life?" In the following sections, we describe the first experiment that provided us with evidence for the existence of such signals in the social web.

4.2.1 Computing a term-based score from Twitter corpus

We compile a set of textual markers $\mathcal{M} = \{m_i\}$, where $i = 1$, 2, ..., k, and look for them in the Twitter corpus of a day. A textual marker can be a single word (1-gram) or a set of words forming a phrase (or n-gram, where n is the number of words). The set of markers $\mathcal{M}$ should point to a target topic – we have chosen an illness, and in particular flu. This choice was not very random; picking flu rates from the social web is an important task that could help in the early detection of epidemics and there is also some ground truth provided from official health authorities, *i.e.* data that could validate the performance of our methodologies.

The daily set of tweets is denoted as $\mathcal{T} = \{t_j\}$, where $j = 1$, 2, ..., n. If a marker m_i appears in a tweet t_j, we set $m_i(t_j) = 1$; otherwise $m_i(t_j) = 0$. The score of a tweet $s(t_j)$ is equal to the number of markers it contains divided by the total number of markers used:

$$s(t_j) = \frac{\sum_i m_i(t_j)}{k}, \tag{4.1}$$

where k denotes the total number of markers. We compute the target topic's score from the daily Twitter corpus $f(\mathcal{T}, \mathcal{M})$, as the sum of all the tweet scores divided by the total number of tweets:

$$f(\mathcal{T}, \mathcal{M}) = \frac{\sum_j s(t_j)}{n} = \frac{\sum_j \sum_i m_i(t_j)}{k \times n}, \tag{4.2}$$

where n denotes the total number of tweets for one day. The number of tweets is not the same for every day, thus it is a necessary choice to normalise the score by dividing with the daily number of tweets in order to retrieve a comparable time series of scores.

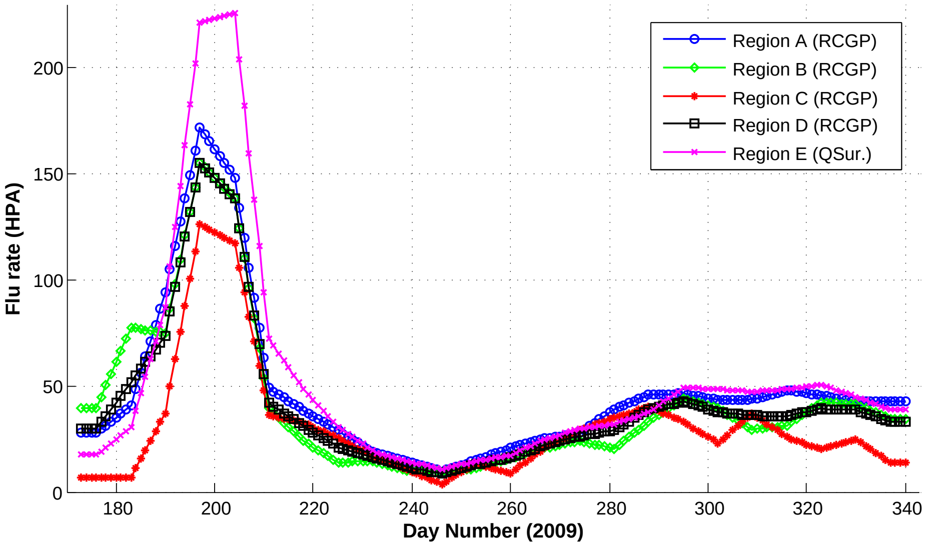


Figure 4.1: Flu rates from the Health Protection Agency (HPA) for regions A-E (weeks 26-49, 2009). The original weekly HPA's flu rates have been expanded and smoothed in order to match with the daily data stream of Twitter.

4.2.2 Correlations between Twitter flu-scores and HPA flu rates

As our ground truth basis, we use weekly reports from the HPA related to ILI figures (see also Section 3.3). HPA provides weekly, regional statistics for the UK, based on rates gathered by the RCGP [46], and the more general QSur scheme. RCGP and QSur metrics express the number of GP consultations per 10^5 citizens, where the result of the diagnosis was ILI.

For this experiment, we are using RCGP data for four UK regions, namely Central England & Wales (region A), South England (region B), North England (region C), and England & Wales (region D), whereas QSur covers England, Wales & Northern Ireland (region E). In order to retrieve an equal representation between the weekly HPA flu rates and the daily Twitter flu-scores, we expand each point of the former over a 7-day period; in fact, each weekly point of an original HPA flu rate is assigned on every day of the respective week. After expanding the HPA flu rates, we perform smoothing on them with a 7-point moving average.[1] Figure 4.1 shows the expanded and smoothed HPA's flu rates for regions A-E during weeks 26-49 in 2009. The early peak in all time series reflects the swine flu (H1N1) epidemic that emerged in the UK during June and July, 2009 [79].

[1] N-point moving average is explained in Appendix A.4.

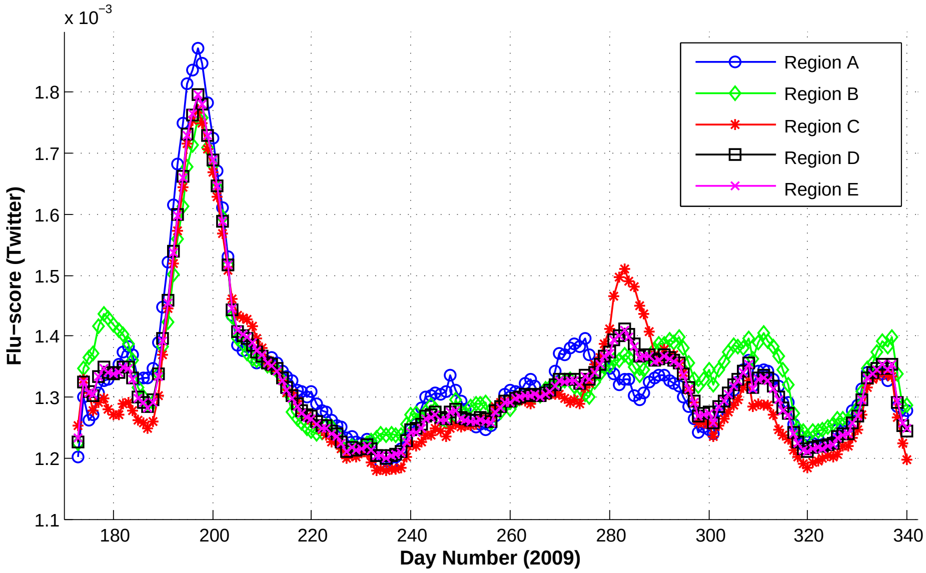


Figure 4.2: Twitter's flu-scores based on our choice of markers for regions A-E (weeks 26-49, 2009). Smoothing with a 7-point moving average (the length of a week) has been applied.

Using a small set of 41 textual markers expressing illness symptoms or relevant terminology, *e.g.* 'fever', 'temperature', 'sore throat', 'infection', 'headache', and so on,[2] we compute the Twitter flu-score time series for regions A–E. We smooth each time series with a 7-point moving average in order to express a weekly tendency in our data. Figure 4.2 shows the time series of Twitter's flu-scores for days 173 to 340 in 2009.

For each one of the five regions, we compute the linear correlation coefficients between Twitter's and HPA's flu-score time series. The correlations and their respective p-values are presented in Table 4.1; the average correlation is equal to 0.8387. The largest correlation of 0.8556 (with a p-value of 2.39e-49) is found for region D, whereas the smallest reported correlation is 0.8178 for region E (with a p-value of 1.1e-41).

Investigating those linear correlations further, we try different window spans for the moving average smoothing function as well as a different approach known as locally weighted scatterplot smoothing (**lowess**). Each smoothed value produced by lowess is essentially given by performing a weighted OLS regression across the points of the smoothing window.[3] The

[2] The entire set of markers is available at Appendix B.2.

[3] Lowess algorithm is presented and thoroughly described in [25]. For our purposes we are using its implementation in MATLAB.

Table 4.1: Linear correlation coefficients between flu-scores derived from Twitter content and HPA's rates based on our choice of markers (for weeks 26-49, 2009) using a 7-point moving average.

	Region	HPA Scheme	Corr. Coef.	P-value
A	Central England & Wales	RCGP	0.8471	1.95e-47
B	South England	RCGP	0.8293	8.37e-44
C	North England	RCGP	0.8438	9.84e-47
D	England & Wales	RCGP	0.8556	2.39e-49
E	England, Wales & N. Ireland	QSur	0.8178	1.10e-41

Table 4.2: Average linear correlation coefficients across all the considered regions between flu-scores derived from Twitter content and HPA's rates for two smoothing functions (n-point moving average and lowess) and various smoothing window sizes. The p-value for all the correlations is < 1.0e-21.

	–	3	5	7	9	11	13	15
***n*-point MA**	0.6933	0.7629	0.8096	0.8387	0.8559	0.8728	0.8863	0.8926
lowess	0.6933	0.6933	0.7576	0.7899	0.8149	0.8356	0.8506	0.8625

results are presented in Table 4.2; we see that increasing the span of the smoothing window produces improved linear correlations under both smoothing functions. Overall moving average smoothing yields better correlations than lowess.

In Figure 4.3, we have plotted the z-scores[4] of Twitter flu score against the corresponding official flu rates from HPA for region D (England & Wales). The high correlation of those two signals can directly be noticed without the need of any statistical measure. This was our first important observation. In the following sections, we explain – in detail – the steps we took in order to actually build on this observation.

4.3 Learning HPA's flu rates from Twitter flu-scores

We extend our previous scheme in order to form a model for predicting a regional HPA flu rate by observing its corresponding flu-score on Twitter. In the new scheme, we attach a

[4] See Appendix A.1 for the definition of a z-score.

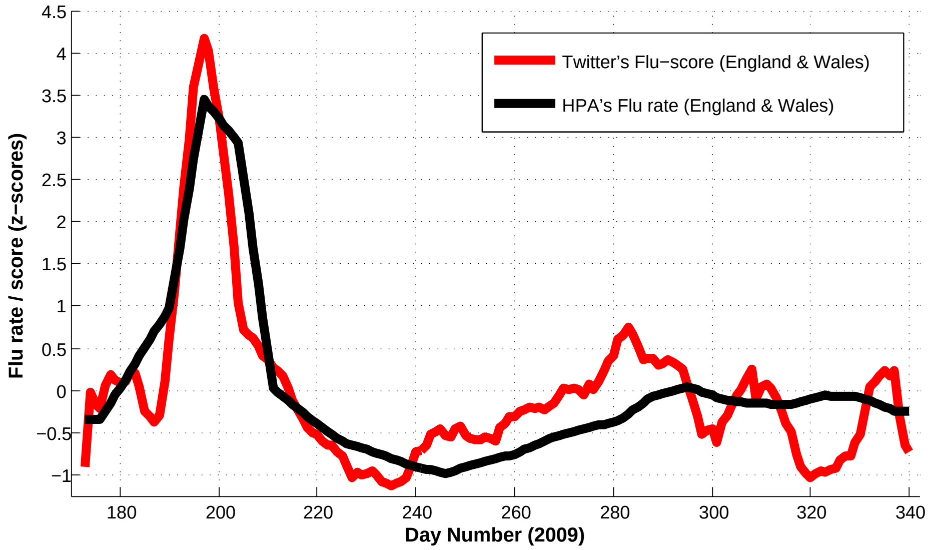


Figure 4.3: Comparison of the unweighted Twitter's flu score (based on our choice of markers) and the respective HPA rates for region D (England & Wales) using their z-scores. Their linear correlation is equal to 0.8556.

weight w_i to each textual marker m_i. The weighted flu-score of a tweet is equal to:

$$s_w(t_j) = \frac{\sum_i w_i \times m_i(t_j)}{k}, \tag{4.3}$$

where k denotes the number of markers. Similarly, the weighted flu-score based on Twitter's daily corpus $\mathcal{T}$ is computed by:

$$f_w(\mathcal{T}, \mathcal{M}) = \frac{\sum_j s_w(t_j)}{n} = \frac{\sum_j \sum_i w_i \times m_i(t_j)}{k \times n}, \tag{4.4}$$

where n denotes the total number of tweets for this day. The contribution of each marker m_i in f_w can be considered as a flu-subscore and is equal to:

$$f_{w_i}(\mathcal{T}, m_i) = w_i \times \frac{\sum_j m_i(t_j)}{k \times n}. \tag{4.5}$$

Therefore, a daily flu-score derived from Twitter content can be represented as a vector $\mathcal{F}_w$ of k elements $\mathcal{F}_w = [f_{w_1}(\mathcal{T}, m_1), ..., f_{w_k}(\mathcal{T}, m_k)]^T$ each one corresponding to Twitter's flu-subscore for marker m_i.

Initially, we retrieve from the Twitter corpus of a day an unweighted flu-score vector $\mathcal{F} =$

$[f(\mathcal{T}, m_1), ..., f(\mathcal{T}, m_k)]^T$. The unweighted time series of each term's flu-subscores ($f(\mathcal{T}, m_i)$ for all the days) is smoothed with a 7-point moving average. We perform OLS regression between the time series of $\mathcal{F}$'s smoothed version and the expanded and smoothed HPA's flu rates in order to learn the weights w_i for the terms m_i. We use as a training set the data that correspond to one region, and then we test the predictability of the inferred weights on the remaining four regions. We perform this training/testing method for all possible (five) training choices.

The linear correlation coefficient between the inferred and the official time series for the HPA flu rates is used as the performance indicator. The results are presented in Table 4.3; the correlation coefficients that were retrieved after training on a region A-E are presented in the row A-E respectively. The average performance over all possible training and testing choices is equal to 0.8994 and the p-values of all the presented correlations indicate strong statistical significance (all of them are $<$ 10e-32). We achieve the maximum average performance (0.9314) when tweets from region A (Central England & Wales) are used for training; the linear correlation of the flu-scores' time series and the HPA flu rates for region E by applying the weights learnt from region A is equal to 0.9723.

Similarly to the previous section, we also report results for various smoothing window spans as well as for the lowess smoothing function (Table 4.4). Interestingly, we see that, under both smoothing methods, correlations reach a maximum for a window of 5 points (*i.e.* days) and then as the smoothing time span increases – especially for the lowess function – performance shows a decreasing behaviour. Overall, lowess does not perform better than moving average; in general, the two smoothing functions produce results of a very similar performance. Based on the facts that: (a) linear correlations for the spans of 5 and 7 do not differ much, (b) using a 7-point moving average produces a more interpretable result as it unveils a weekly tendency from the data, and (c) moving average performs slightly better than lowess, in the results presented in the following sections, we only perform smoothing using a 7-point moving average, leaving further investigation of smoothing spans and methodologies to future work.

To assess the predictive power of the former result differently, we perform linear regression on the aggregated time series of the flu-scores and HPA's flu rates using the data from all the regions. The data belonging in weeks 28 and 41 (during the peak and the stabilised period of the epidemic respectively) form the test set; the remaining data is used for training the

Table 4.3: Linear regression using the markers of our choice - An element (i, j) denotes the correlation coefficient between the weighted flu-scores time series and HPA's flu rates on region j, after training the weights on region i. The p-value for all the correlations is $<$ 10e-32.

Train/Test	**A**	**B**	**C**	**D**	**E**	Avg.
A	-	0.8389	0.9605	0.9539	0.9723	**0.9314**
B	0.7669	-	0.8913	0.9487	0.8896	0.8741
C	0.8532	0.702	-	0.8887	0.9445	0.8471
D	0.8929	0.9183	0.9388	-	0.9749	0.9312
E	0.9274	0.8307	0.9204	0.9749	-	0.9134
				Total Avg.		**0.8994**

Table 4.4: Average linear correlation coefficients across all the considered regression scenarios between the inferred flu-scores derived from Twitter content and HPA's rates for two smoothing functions (n-point moving average and lowess) and various smoothing window sizes. The p-value for all the correlations is $<$ 1.0e-11.

	–	**3**	**5**	**7**	**9**	**11**	**13**	**15**
***n*-point MA**	0.8849	0.9078	0.9101	0.8994	0.9039	0.9055	0.8618	0.8152
lowess	0.8849	0.8849	0.9072	0.9064	0.9002	0.8898	0.8627	0.83

weights. This results to a linear correlation of 0.9234 with a p-value of 5.61e-30 on the test set. Additionally, we perform a 10-fold cross validation (1000 repeats, where the folds are randomly decided each time) using again linear regression for learning. On average, we obtain a linear correlation of 0.9412 with a standard deviation equal to 0.0154.

While in all cases the score was tested on unseen data, in the first set of experiments we trained on data gathered on one region, and then tested on the remaining regions, but on the same period of time; in the last two experiments, using an aggregation of our data sets, we carried out training and testing on different times. Together these two sets of experiments, provide strong support to the predictive power of the flu-score we developed.

4.3.1 What is missing?

In the previous sections, we used a set of handpicked markers related to the target topic of illness to infer ILI scores in several regions in the UK. Obviously, this is not an optimal choice; how can one know that the optimal set of markers is being used, especially when she deals with a vague concept? In the following sections, we show ways to select those features automatically, enabling our approach to be general and independent of human effort and domain knowledge.

4.4 Automatic extraction of ILI textual markers

In the previous sections, we made use of hand crafted ILI related textual markers. In this section, we present a method for extracting weighted markers (or features) automatically. The method selects a subset of keywords and their weights to maximise the correlation with the HPA flu rates, while also minimising the size of the keyword set. It is formed by two parts: creating a set of candidate features, and then selecting the most informative ones.

At first, we create a pool of candidate markers from web articles related to influenza. We use an encyclopedic reference[5] as well as a more informal reference where potential flu patients discuss their personal experiences.[6] After preprocessing (tokenisation, stop-word removal), we extract a set of $K = 1560$ stemmed candidate markers (1-grams). The latter is denoted by $\mathcal{M}_C = \{m_{ci}\}$, $i \in [1, K]$. $\mathcal{M}_C$ contains some words which form a very good description of the topic as well as many irrelevant ones. This formation choice for the set of candidate markers is discussed further and justified in Sections 4.5 and 4.6.

After forming the candidate features, we compute their daily, regional, and unweighted flu-subscores $f(\mathcal{T}_r, m_{ci})$ given $\mathcal{T}_r$ which denotes the Twitter corpus for region r, $r \in \{$A-E$\}$. For a day d, the flu score on Twitter is represented as a vector $\mathcal{F}_{d,r} = [f(\mathcal{T}_r, m_{c1}) \ ... \ f(\mathcal{T}_r, m_{cK})]^T$. Consequently, for a region r and a period of ℓ days, we can form an array with the time series of the flu-subscores for all the candidate features: $X_r = [\mathcal{F}_{1,r} \ ... \ \mathcal{F}_{\ell,r}]^T$, where ℓ denotes the total number of days considered. The columns of X_r, *i.e.* the time series of the flu-subscores of each candidate feature, are smoothed using a 7-point moving average (as in the previous cases); the resulting array is denoted as $X_r^{(s)}$.

The expanded and smoothed time series of the HPA's flu rates for region r and for the same period of ℓ days are denoted by the vector $h_r^{(s)}$. At this point, one could use the correlation

[5] Influenza on Wikipedia, http://en.wikipedia.org/wiki/Influenza.

[6] Swine Flu on NHS (with potential patients comments), http://www.nhs.uk/Conditions/pandemic-flu/Pages/Symptoms.aspx.

coefficient between each column of $X_r^{(s)}$ and $h_r^{(s)}$ or other linear regression methods (least squares, rigde regression, etc.) in order to rank or learn weights for the candidate features. For this purpose, the LASSO method has been chosen as it has the advantage of producing sparse solutions, *i.e.* it will discard candidate features which are proven to be redundant in terms of predictability [158]. LASSO is an established method for estimating least squares parameters subject to an L1 penalty. It can be considered as a constrained optimisation task, which in our case is formulated as

$$\begin{aligned} \min_{w} \quad & \|X_r^{(s)}w - h_r^{(s)}\|_2^2 \\ \text{s.t.} \quad & \|w\|_1 \leq t, \end{aligned} \tag{4.6}$$

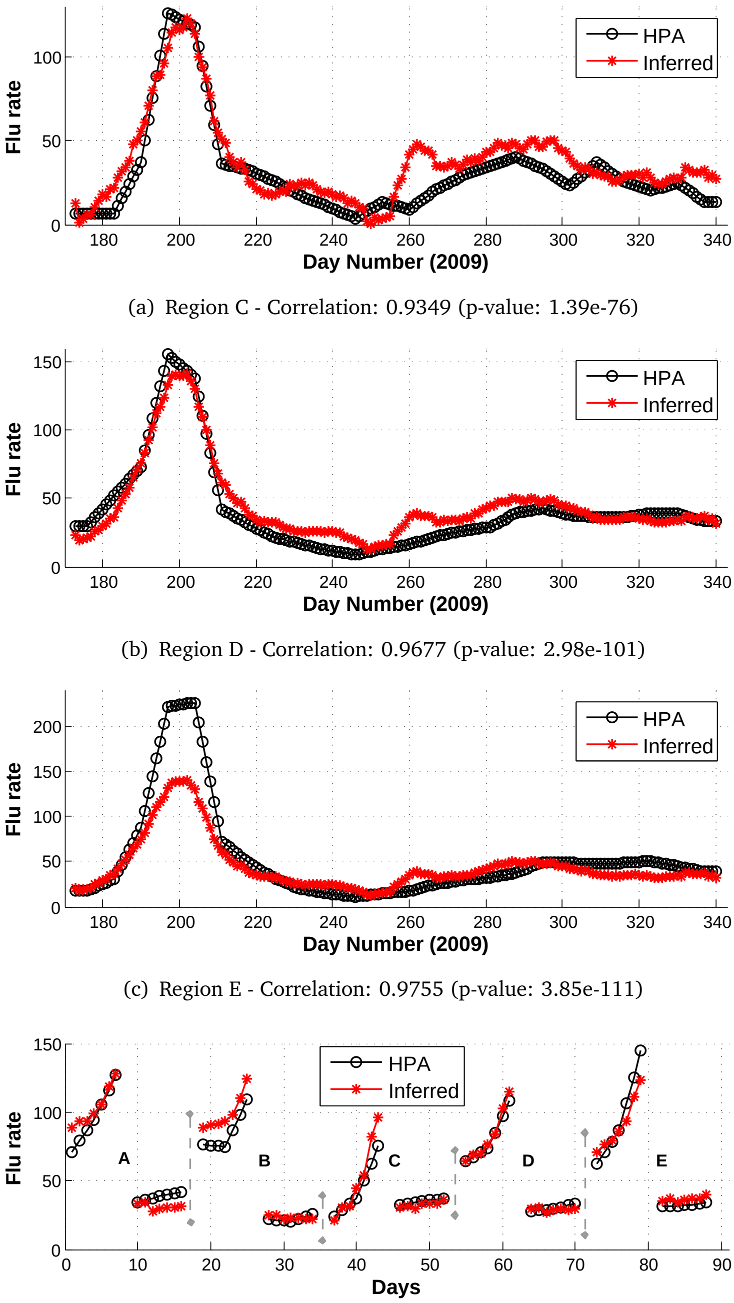
where vector w is the sparse solution, and t is the shrinkage parameter. The shrinkage parameter can be expressed as

$$t = \alpha \times \|w^{(ls)}\|_1, \tag{4.7}$$

where $w^{(ls)}$ denotes the least squares estimates for our regression problem, and $\alpha \in (0, 1)$ is the shrinkage percentage.

We use time series of a region $r_i \in$ {A-E} as the training set, the time series of a region $r_j \in$ {{A-E} $- r_i$} as the validation set for deciding the optimal shrinkage percentage α, and we test on the data of the remaining three regions. We repeat this procedure for all possible five training set choices. LARS algorithm is applied to compute LASSO's estimates [39]. The results of our method are captured in Table 4.6. Most of the possible training/validating choices lead to high linear correlations. The average linear correlation over all possible settings is 0.9256 indicating the robustness of our method. The experiments showed that the optimal scenario was to train on region A and use region B for validating α, leading to an average correlation of 0.9594 on the remaining three regions (C-E) (for a shrinkage percentage α equal to 87%). Figures 4.4(a), 4.4(b), and 4.4(c) show a comparison between the inferred and HPA's flu rates time series on regions C-E respectively (for the optimal choice). The weight vector w had 97 non-zero values, *i.e.* we were able to extract 97 markers (or features), which, in turn, are presented in Table 4.5. The majority of the markers is pointing directly or indirectly to illness related vocabulary.

We also assess the performance of our method differently, following the same principle as in the previous section. We aggregate our regional data sets $X_r^{(s)}$ and $h_r^{(s)}$, and as before, we form a test set by using the data for weeks 28 and 41, a validation set (for deciding the



(a) Region C - Correlation: 0.9349 (p-value: 1.39e-76)

(b) Region D - Correlation: 0.9677 (p-value: 2.98e-101)

(c) Region E - Correlation: 0.9755 (p-value: 3.85e-111)

(d) Inference on the aggregated data set (per region) for weeks 28 and 41 - Correlation: 0.9713 (p-value: 3.96e-44)

Figure 4.4: Comparison of the inferred versus the HPA's flu rates, after using LASSO method for learning. Figures 4.4(a), 4.4(b), and 4.4(c) present the results for the regional experiment (training on one region, validating on another, and testing on the remaining three), whereas 4.4(d) shows the results for the aggregated sets (5 partitions of a 14-day period each from left to right correspond to the results for regions A-E respectively). Note that for subfigure 4.4(d) the 'Days' axis does not denote yearly day numbers.

Table 4.5: 97 stemmed markers extracted by applying LASSO regionally. The markers are sorted in a descending order based on their weights (read horizontally, starting from the top-left corner).

lung	unwel	temperatur	like	headach	season
unusu	chronic	child	dai	appetit	stai
symptom	spread	diarrhoea	start	muscl	weaken
immun	feel	liver	plenti	antivir	follow
sore	peopl	nation	small	pandem	pregnant
thermomet	bed	loss	heart	mention	condit
high	group	tired	import	risk	carefulli
work	short	stage	page	diseas	recognis
servic	wors	case	similar	term	home
increas	exist	ill	sens	counter	better
cough	vomit	earli	neurolog	catch	onlin
fever	concern	check	drink	long	far
consid	ach	breath	flu	member	kidnei
mild	number	sick	throat	famili	water
read	includ	swine	confirm	need	nose
medic	phone	cancer	disord	unsur	suddenli
runni					

Table 4.6: Linear correlations on the test sets after performing the LASSO - An element (i, j) denotes the average correlation coefficient on the three remaining regions, after performing LASSO on region i in order to learn the weights, and validating the shrinkage parameter t on region j.

Train/Validate	**A**	**B**	**C**	**D**	**E**
A	-	**0.9594**	0.9375	0.9348	0.9297
B	0.9455	-	0.9476	0.9267	0.9003
C	0.9154	0.9513	-	0.8188	0.908
D	0.9463	0.9459	0.9424	-	0.9337
E	0.8798	0.9506	0.9455	0.8935	-
				Total Avg.	**0.9256**

optimal value of the shrinkage percentage α) by using weeks 36 and 49, and a training set with the remaining data sets. The outcome of the experiment indicates that the optimal

Table 4.7: 73 stemmed markers extracted by applying LASSO on the aggregated data set of regions A-E. The markers are sorted in a descending order based on their weights (read horizontally, starting from the top-left corner). 72 of them have also occurred in Table 4.5 but here the order (*i.e.* the corresponding weights) is different.

muscl	like	appetit	read	unwel	child
work	follow	season	page	throat	nose
check	suddenli	pleas	immun	phone	swine
sick	dai	symptom	consid	sens	breath
cough	loss	recognis	peopl	number	mild
home	condit	mention	servic	runni	member
wors	diseas	diarrhoea	high	short	onlin
pregnant	small	exist	headach	unsur	cancer
stai	concern	fever	earli	tired	carefulli
import	weaken	nation	famili	similar	temperatur
feel	ach	flu	case	sore	unusu
spread	vomit	ill	thermomet	pandem	increas
stage	far				

value for α is 0.0049; for this value we retrieve a linear correlation of 0.9713 (p-value is equal to 3.96e-44) on the test set. The corresponding vector of weights w has 73 non-zero features which are shown in Table 4.7. Only one of them (the stemmed word ‘pleas’) was not included in the previously extracted set of features. Figure 4.4(d) presents a comparison of the inferred versus the HPA’s flu rates for all the test points. Again, we have demonstrated how a list of markers can be automatically inferred from a large set of candidates by using a supervised learning algorithm and HPA’s index as the target signal; this approach delivers a correlation greater than 0.97 with the target signal on unseen data.

4.4.1 What is missing from this approach?

We have showcased how a sparse learning method can be applied in order to automatically select flu-related markers and then use them to track the levels of flu in several UK regions. In this experiment, we have chosen to smooth the input data (the time series of each marker’s frequency) with a 7-point moving average to express a weekly tendency – however, when applying this method on new unseen data, those data cannot or should not always be smoothed. Furthermore, the use of single words only (1-grams) might favour overfitting; single words unrelated with the target topic have a larger probability to correlate with it by chance than larger chunks of text [52, 157]. Moreover, LASSO has known disadvantages (see Section 2.2.3) and especially is inconsistent in selecting features. Finally, is this approach general? Will this methodology work when applied on other topics as well? We try to address those issues as well as others (see Section 5.1) in the next chapter.

4.5 The Harry Potter effect

A mainstream IR technique (for example the VSM, see Section 2.6.1) proposes the formation of an index or vocabulary using all the n-grams in the target corpus [5, 83, 104]. In our approach, however, we form an index from several encyclopedic or other topic related web references, a choice that is partly justified in this as well as in the next section.

Following this basic principle of the VSM, for weeks 26–49 in 2009 (the same 168-day time period used in the previously presented experimental process), we index all 1-gram, after applying Porter stemming and removing stop words that appear more than 50 times in the Twitter corpus for England & Wales. The retrieved index contains 47,193 terms. The time series of each 1-grams is smoothed by applying a 7-point moving average; the weekly ground truth is expanded and smoothed.

At first, in order to get a quick characterisation of the data set, we compute the linear correlation of each 1-gram with the ground truth. Table 4.8 shows the top-20 correlated markers together with their correlation coefficients (all correlations are statistically significant). There exist some flu related words such as ‘flu’ or ‘swine’ but the majority of the words is unrelated to the topic of flu and seems to derive from the well known Harry Potter fantasy novel/film series. The top correlated word is ‘latitud’, something that at first sight might not make much sense.

Knowing that in the region of England & Wales the flu epidemic peaked on the 16th of July and basing our search on the top correlated terms (Table 4.8), we tracked events that emerged at the same date and might have been a topic of conversation on Twitter. Indeed, a Harry Potter movie aired on the 15th of July in the UK.[7] Moreover, two 3-day music

[7] According to the Internet Movie Database – http://www.imdb.com/title/tt0417741/releaseinfo.

Table 4.8: Top-20 linearly correlated terms with the flu rate in England & Wales

1-gram	Event	Corr. Coef.
latitud	Latitude Festival	0.9367
flu	Flu epidemic	0.9344
swine	▲	0.9212
harri	Harry Potter Movie	0.9112
slytherin	▲	0.9094
potter	▲	0.8972
benicassim	Benicàssim Festival	0.8966
graduat	Graduation (?)	0.8965
dumbledor	Harry Potter Movie	0.8870
hogwart	▲	0.8852
quarantin	Flu epidemic	0.8822
gryffindor	Harry Potter Movie	0.8813
ravenclaw	▲	0.8738
princ	▲	0.8635
swineflu	Flu epidemic	0.8633
ginni	Harry Potter Movie	0.8620
weaslei	▲	0.8581
hermion	▲	0.8540
draco	▲	0.8533
snape	▲	0.8486

festivals kicked off on July 16th: the annual festival ‘Latitude’ featuring several popular artists[8] and an annual international festival in Spain (‘Bennicàssim’) – possibly popular among UK citizens – which was interrupted by a large fire.[9]

This, conventionally named as the **Harry Potter effect**, is something expected to happen when other signals emerging from real-life events show similar behaviour compared to the target topic (see Figure 4.5). The possibility of having other events in correlation with the

[8] According to the relevant article on Wikipedia, Latitude Festival – http://goo.gl/2oQmF.

[9] According to the relevant article on Wikipedia, Festival Internacional de Benicàssim – http://goo.gl/jfHKk.

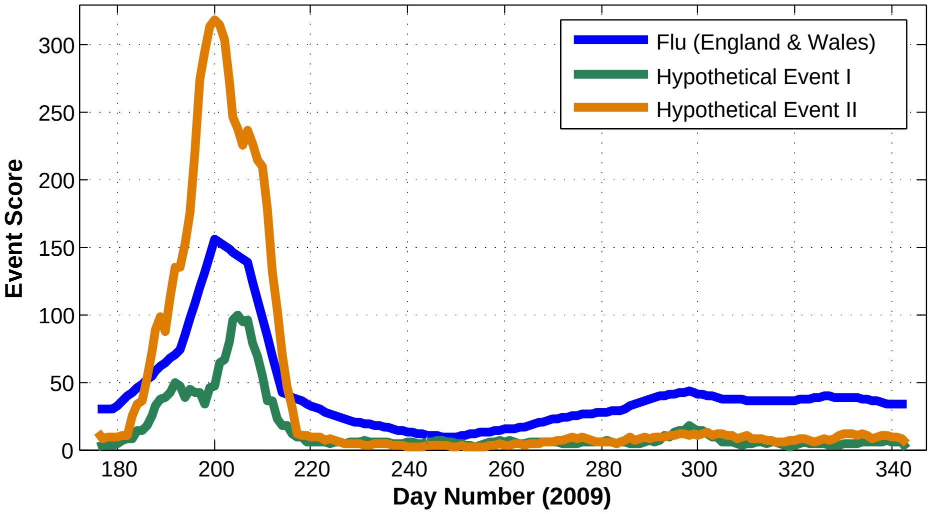


Figure 4.5: The Harry Potter effect - There might exist other signals emerging from real-life events correlated with the target topic.

target one is increasing when the data are from a relatively small period of time and it is even higher when the target event emerged only once in this period.

We have also applied LASSO on the same data set, knowing that it might not select the top correlated terms but a different set of variables conforming with its constraints. After permuting the data set based on a random selection of the day-index, we perform learning by applying a 5-fold cross validation scheme. The performance of the selected features and their weights on the disjoint test set is excellent, *i.e.* of the same level as in Section 4.4. However, by observing the selected features and their weights in every iteration of the process, we have proof that this is a case of overfitting. Table 4.9 shows the top-20 selected terms (based on their weights) for all steps of the cross validation process. The selected features are slightly different in every iteration, something expected as they are based on training sets that are 20% different from one another. Again, a few basic flu-related features have been selected, such as ‘flu’ and ‘swine’ as well as the word ‘harri’. However, most terms seem to describe other events (which we did not track), such as a lottery or a graduation ceremony.

With a minority of features related to the target topic of illness, it is safe to conclude that any significant performance on the test set is a result of overfitting. Bearing in mind this ‘Harry Potter effect’, we chose to form the set of candidate markers in a more focused way as it has been described in the previous section.

Table 4.9: Top-20 selected terms by LASSO (sorted in descending order based on their weight) in every iteration of the 5-fold cross validation.

it-**1**	it-**2**	it-**3**	it-**4**	it-**5**
swine	hetero	flu	graduat	flu
thunder	khan	khan	flu	graduat
graduat	flu	swine	hetero	hetero
fern	harri	bastil	harri	media
harri	lotteri	fern	swine	harri
sale	fern	harri	sundai	swine
flu	swine	ladi	manchest	thunder
market	meteor	manchest	school	sundai
hetero	chrome	mobil	weekend	manchest
mondai	memori	sale	meteor	meteor
sundai	player	app	god	school
labour	feel	servic	live	record
gener	sundai	quiz	app	gener
bristol	saturdai	market	lotteri	wood
prize	help	mondai	moir	death
public	tori	week	player	footbal
khan	edward	level	memori	canada
lotteri	ash	danyl	andi	spammer
award	xxx	bank	bed	memori
clock	wave	graduat	feel	award

4.6 How error bounds for L1-regularised regression affect our approach

Suppose that we have an input matrix X with N samples and p variables (or dimensions), such that $x_{ij} \in X$ with $i \in \{1, ..., N\}$ and $j \in \{1, ..., p\}$, and an one dimensional target variable y of size N. The empirical LASSO estimate $\hat{w}$ is given by:

$$\hat{w} = \arg\min_{w} \left\{ \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{p} x_{ij} w_j \right)^2 + \lambda \sum_{j=1}^{p} |w_j| \right\}, \tag{4.8}$$

where λ is the shrinkage parameter.

It is important to notice that statistical bounds exist linking LASSO's expected performance to the one derived on the training set (empirical), the number of dimensions, number of training samples and L1-norm or sparsity of $\hat{w}$.

A general error bound drawn on the Euclidean distance between the optimal and the empirical LASSO coefficients suggests that

$$\|w - \hat{w}\|_{\ell_2} = \mathcal{O}(\sqrt{\frac{k \log p}{N}}), \tag{4.9}$$

where k is the number of nonzero elements in $\hat{w}$ [168]. Equation 4.9 makes clear that the error is proportional to the dimensionality of the problem, but disproportional to the number of samples or the achieved sparsity of $\hat{w}$.

In [7] it is shown that LASSO's expected loss $\mathcal{L}(w)$ up to poly-logarithmic factors in W_1, p and N is bounded by

$$\mathcal{L}(w) \leq \mathcal{L}(\hat{w}) + \mathcal{Q}, \text{ with } \mathcal{Q} \sim \min\left\{\frac{W_1^2}{N} + \frac{p}{N}, \frac{W_1^2}{N} + \frac{W_1}{\sqrt{N}}\right\}, \tag{4.10}$$

where $\mathcal{L}(\hat{w})$ denotes the empirical loss and W_1 is an upper bound for the L1-norm of $\hat{w}$, *i.e.* $\|\hat{w}\|_{\ell_1} \leq W_1$; this error bound is sharp (not loose) and predictive. Therefore, to minimise the prediction error using a fixed set of training samples and given that the empirical error is relatively small, one should either reduce the dimensionality of the problem (p) or increase the shrinkage level of $\hat{w}$'s L1-norm (which intuitively might result in sparser solutions, *i.e.* fewer selected features).

A more complex error bound regarding the predictions (Xw), derived in [23] and holding with probability at least $1 - 6p^{-2\log 2} - p^{-1}(2\pi \log p)^{-1/2}$ is the following:

$$\|Xw - X\hat{w}\|_{\ell_2}^2 \leq C_0 \cdot (2 \log p) \cdot k \cdot \sigma^2, \tag{4.11}$$

where $C_0 = 8(1 + \sqrt{2})^2$, k is the number of nonzero elements in $\hat{w}$ and the prediction errors ϵ_i, $j \in \{1, ..., N\}$ are assumed to be Gaussian with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. For this equation to hold, we also need an X with no highly co-linear predictors.

In all the aforementioned error bounds, as well as in many others in literature [9, 59, 82], the main conclusion is similar: as the dimensionality of the problem increases, error increases as well. When p/N (dimensionality – sample-size ratio) is relatively small, *i.e.* when the number of variables is far less than the number of samples, the error rate is reduced. Which, in turn, means that when the number of samples is relatively small, in order to achieve a satisfactory inference performance, one has to constrain the number of dimensions; this serves as an additional reasoning for our feature extraction modelling choice. Finally, we see that sparser solutions result to lower error rates – which was one of the reasons for using LASSO in the first place.

4.7 Summary of the chapter

In this chapter, we have presented a preliminary method for tracking the flu epidemic in the UK by using the contents of Twitter; our approach could give an early warning in various situations, but mostly can give timely and free information to health agencies to plan health care. This method is based on the textual analysis of micro-blog contents, is calibrated with the ground truth provided by the HPA, and could also be extended to the contents of text messages sent via mobile devices (besides privacy concerns), as the latter are of the same format with tweets (140 characters limitation).

At first, we formed a set of keywords or phrases which represented illness and some of the flu symptoms. After defining a basic function which combined the normalised frequencies of those manually selected terms to come up with a flu-score, we discovered that the time series of those flu scores were linearly correlated with the actual flu rates. By adding one more layer to this approach, we applied OLS regression to learn a weight for each term; as expected, the retrieved correlations between OLS predictions and the ground truth were greater than the original ones. The next step included the automatic selection of features from the textual stream. To achieve this we applied a sparse regressor, the LASSO, which was able to select a small, weighted subset of the initial feature space; the selected features were – in their great majority – drawn from the illness topic. This automatic method, improved the inference performance further. However, the original set of candidate features was not formed by using the entirety of the Twitter corpus, something that is usually the mainstream approach in IR. We have chosen to form the candidate features from text which was related with the target topic such as Wikipedia pages and discussions from health-oriented websites; nevertheless, the majority of the candidate features were irrelevant to the illness topic. This choice is justified firstly by presenting and explaining the ‘Harry Potter effect’ in event detection and secondly by understanding how error bounds for LASSO are affected by the specific characteristics of a learning task.

The approach presented in this chapter can automatically learn the inputs (textual markers and their weights) of a scoring function that correlates very highly with the HPA flu score. The proposed method only requires access to the time series of geolocated blog contents and a ground truth. However, knowing that LASSO is an inconsistent learning function in terms of model selection, but also that the presented method was topic-specific, in the next chapter we focus on improving it in several aspects as well as making it – to an extent – generic.

Chapter 5

Nowcasting Events from the Social Web with Statistical Learning

We present a generic methodology – which builds on the methodology presented in Chapter 4 resolving some of its limitations – for inferring the occurrence and magnitude of an event or phenomenon by exploring the rich amount of unstructured textual information on the social part of the web. Having geo-tagged user posts on the microblogging service of Twitter as our input data, we investigate two case studies. The first consists of a benchmark problem, where actual levels of rainfall in a given location and time are inferred from the content of tweets. The second one is a real-life task, where we infer regional ILI rates in the effort of detecting timely an emerging epidemic disease. Our analysis builds on a statistical learning framework, which performs sparse learning via the bootstrapped version of LASSO to select a consistent subset of textual features from a large amount of candidates. In both case studies, selected features indicate close semantic correlation with the target topics and inference, conducted by regression, has a significant performance; the performance of our method is also compared with another approach proposed to address a similar task, showing a significant improvement. We also apply a different, nonlinear and non-parametric learner – an ensemble of CARTs – in the core of our approach, and investigate possible differences in the feature selection and inference process. This chapter is an extended version of our paper "Nowcasting Events from the Social Web with Statistical Learning" [91].

5.1 Recap on previous method's limitations

We have already described in Chapter 4 a preliminary method which enabled us to detect flu-like illness in several UK regions by observing Twitter content. However, the core theoretical tool used in this approach, a sparse L1-regulariser known as the LASSO, has been proven to be a model-inconsistent learner as in many settings it selects more variables than necessary [103, 173]. Taking a closer look at the proposed method, one could also argue that the use of 1-grams (unigrams) only could also lead to a bad choice of features and consequently to overfitting. Based on the simple notion that the entropy – or the uncertainty – of the English language decreases as the number of characters increases [147], and considering the experimental proof for the learning improvements which may be achieved by the use of 2-grams (bigrams) [52, 157], in this chapter we present a hybrid scheme that combines unigrams and bigrams. Furthermore, the previous experimental process has been focused on a short time span (168 days), and there was no comparison with other methods proposed for similar tasks. One more critical point was the use of smoothing on the input data, something that improved the performance of our approach. However, in real-time applications, where inferences should be made about the running state of a hidden variable, this type of smoothing might cause the loss of important information. Lastly, our previous experiments were focused on a single specific event (Swine Flu epidemic in the UK, 2009) and therefore, no experimental proof existed allowing us to generalise our findings.

5.2 Nowcasting events

The term '*nowcasting*', commonly used in economics [6, 55] as well as in meteorology [108, 170], expresses the fact that we are making inferences regarding the current magnitude $\mathcal{M}(\varepsilon)$ of an event ε – in this special case 'current' may refer to the very near future, but also the very recent past.

For a time interval $u = [t - \Delta t, t + \Delta t]$, where t is a reference time and $(t + \Delta t)$ denotes the current time instance, consider $\mathcal{M}\left(\varepsilon^{(u)}\right)$ – the magnitude of an event ε within the time interval u – as a latent variable. For the same time instance u, we denote the entire web content as $\mathcal{W}^{(u)}$ and an observable proportion of it as $\mathcal{S}^{(u)}$, where $\mathcal{S}^{(u)} \subseteq \mathcal{W}^{(u)}$.

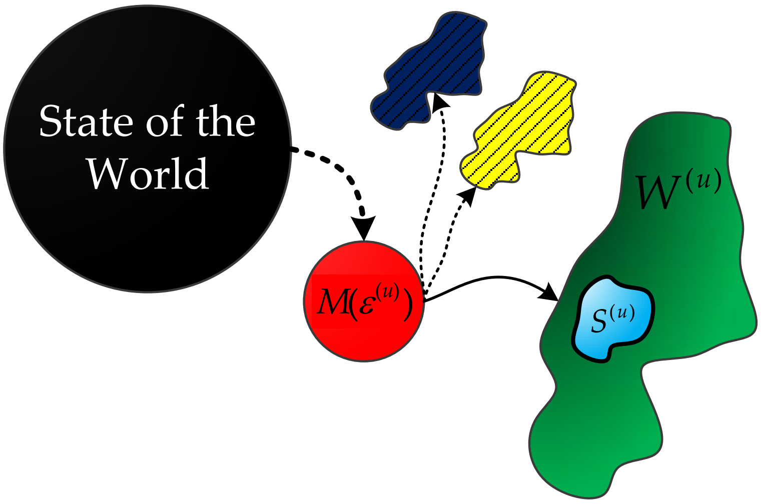


Figure 5.1: An abstract graphical representation that describes the generic hypothesis behind our model. The State of the World during a time interval u affects the occurrence and magnitude of an event ε, $\mathcal{M}\left(\varepsilon^{(u)}\right)$. In turn, this event might affect other entities including the content of the Web, $\mathcal{W}^{(u)}$. We are observing a subset of the Web content, $\mathcal{S}^{(u)}$.

Making the hypothesis that an event emerging in real life will affect the content of the web, our aim is to use the observable web content in order to detect and also infer the magnitude of this event. Using the introduced notation, we want to infer $\mathcal{M}\left(\varepsilon^{(u)}\right)$ directly from $\mathcal{S}^{(u)}$. Given that u is of short length, we are inferring the present value of the latent variable, *i.e.* we are **nowcasting** $\mathcal{M}\left(\varepsilon^{(u)}\right)$.

From a probabilistic point of view (see an abstract graphical representation in Figure 5.1), we first have to learn the conditional probability distribution $\Pr\left(\mathcal{S}^{(u)}|\mathcal{M}(\varepsilon^{(u)})\right)$ from the observable information, and then – following the Bayesian logic – we use $\Pr\left(\mathcal{S}^{(u)}|\mathcal{M}(\varepsilon^{(u)})\right)$ to infer $\Pr\left(\mathcal{M}(\varepsilon^{(u)})|\mathcal{S}^{(u)}\right)$. An important assumption here is that during the time interval u, there was enough time for the web to depict this real-life situation in its content. When u is of a relatively small value, this might not be true for the majority of the web sites, but has been shown to hold for social networks, such as Twitter [139]. Obviously, an even more fundamental assumption made is that an identifiable – possibly causal – link between the shape of the web content and real-life events does exist; indeed, we have already presented results that support this assumption in the previous chapter.

5.3 Related work

Recent work has been concentrated on exploiting user generated web content for conducting several types of inference. A significant subset of papers, examples of which are given in this paragraph, focuses on methodologies that are based either on manually selected textual features related to a latent event, *e.g.* flu related keywords, or the application of sentiment or mood analysis, which in turn implies the use of predefined vocabularies, where words or phrases have been mapped to sentiment or mood scores [122]. Corley *et al.* reported a 76.7% correlation between official ILI rates and the frequency of certain hand-picked influenza related words in blog posts [29], whereas similar correlations were shown between user search queries which included illness related words and CDC[1] rates [127]. Furthermore, sentiment analysis has been applied in the effort of extracting voting intentions [164] or box-office revenues [2] from Twitter content. Similarly, mood analysis combined with a non linear regression model derived a 0.876 correlation with daily changes in Dow Jones Industrial Average closing values [11]. Finally, Sakaki *et al.* presented a method which exploited the content, time stamp and location of a tweet to detect the existence of an earthquake [139].

However, in other approaches feature selection is performed automatically by applying statistical learning methods. Apart from the obvious advantage of reducing human involvement to a minimum, those methods tend to have an improved inference performance as they are enabled to explore the entire feature space or, in general, a greater amount of candidate features [63]. In [56], Google researchers proposed a model able to automatically select flu related user search queries, which later on were used in the process of tracking ILI rates. Their method, a core component of Google Flu Trends, achieved an average correlation of 0.9 with CDC data, much higher than any other previously reported method. An extension of this approach has been applied on Twitter data achieving a 78% correlation with CDC rates [31]. In both those works, features were selected based on their individual correlation with ILI rates; the subset of candidate features (user search queries or keywords) appearing to independently have the highest linear correlations with the target values formed the result of feature selection. Another technique – part of our preliminary results presented in the previous chapter –, which applied sparse regression on Twitter content for automatic feature selection, resulted to a greater than 90% correlation with HPA's flu rates for several

[1] Centers for Disease Control and Prevention (CDC), http://www.cdc.gov/.

UK regions [92] – an improved version of this methodology has been incorporated in Flu Detector [93], an online tool for inferring flu rates based on tweets (see Chapter 8).

Besides the important differences regarding the information extraction and retrieval techniques or the data sets considered, the fundamental distinction between [31, 56] and [92] lies on the feature selection principle – a sparse regressor, such as LASSO, does not handle each candidate feature independently but searches for a subset of features which satisfies its constraints [158] (see also Section 2.2.3).

5.4 Methodology

In this section, we describe the proposed methodology in detail. An abstract summary of it would include the following main operations:

1. A **pipeline** that is used to collect and store the essential observations or input data in general; in our experiments this is Twitter content (and we have already described this process in Chapter 3).
2. **Candidate Feature Extraction**. A vocabulary of candidate features is formed by using n-grams, *i.e.* phrases with n tokens. We also refer to those n-grams as markers. Markers are extracted from text which is expected to contain topic-related words, *e.g.* web encyclopedias as well as other more informal references. By construction the set of extracted candidates contains many features relevant with the target topic and much more with no direct semantic connection. We are considering two schemes of features: 1-grams and 2-grams. Then, we investigate ways to combine them.
3. **Feature Selection**. A subset of the candidate features is selected by applying a sparse learning method.
4. **Inference and Validation**. Finally, we test our proposed method on unseen data and retrieve performance figures.

5.4.1 Forming vector space representations of text streams

We define a text stream as a superset of documents. The word stream denotes that there is a continuous flow of documents given that the Web does not have a static behaviour. Documents can be tweets, status updates on Facebook, blog posts and so on as well as combinations of the aforementioned examples. As we have already mentioned, they are comprised by sets of words known as n-grams, where n indicates the number of words in the set.

In our effort to extract useful signals from textual streams, some n-grams are more informative than others depending always on the task at hand. Our first step towards achieving our goal, which is to detect and quantify such signals, is to spot the most informative n-grams or at least a subset of them which guarantees a significant performance during inference.

All n-grams that take part in this 'selection' process are denoted as candidate n-grams or features or markers. Suppose we have extracted a set of candidate features $\mathcal{C} = \{c_i\}$, $i \in \{1, ..., |\mathcal{C}|\}$.

The user generated textual stream for a time interval u is denoted by the set of posts $\mathcal{P}^{(u)} = \{p_j\}$, $j \in \{1, ..., |\mathcal{P}^{(u)}|\}$.

A function g indicates the frequency φ (raw count) of a candidate marker c_i in a user post p_j:

$$g(c_i, p_j) = \begin{cases} \varphi & \text{if } c_i \in p_j, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

Given the limitation of a tweet's length (140 characters), when we deal with Twitter content, we choose to restrict function g to its Boolean special case by setting the maximum value for φ to 1.

We compute the score s of a candidate marker c_i in the set of user posts $\mathcal{P}^{(u)}$ as follows:

$$s(c_i, \mathcal{P}^{(u)}) = \frac{\sum\limits_{j=1}^{|\mathcal{P}^{(u)}|} g(c_i, p_j)}{|\mathcal{P}^{(u)}|}. \tag{5.2}$$

Therefore, the score of a candidate marker is a normalised frequency representing the ratio of its occurrences over the total number of posts in the text stream. When g is a Boolean function this score can be also interpreted as the ratio of tweets containing the marker over the total number of tweets.

The scores of all candidate markers for the same time interval u are kept in a vector x given by:

$$x^{(u)} = [s(c_1, \mathcal{P}^{(u)}) \text{ ... } s(c_{|\mathcal{C}|}, \mathcal{P}^{(u)})]^T. \tag{5.3}$$

The length of the time interval u is a matter of choice and can vary depending on the inference task at hand. In our work, we frequently set it equal to the duration of a day.

For a set of time intervals $\mathcal{U} = \{u_k\}$, $k \in \{1, ..., |\mathcal{U}|\}$ and given $\mathcal{P}^{(u_k)}$ $\forall k$, we compute the scores of the candidate markers $\mathcal{C}$. Those are held in a $|\mathcal{U}| \times |\mathcal{C}|$ array $\mathcal{X}^{(\mathcal{U})}$:

$$\mathcal{X}^{(\mathcal{U})} = [x^{(u_1)} \text{ ... } x^{(u_{|\mathcal{U}|})}]^T. \tag{5.4}$$

In most occasions time intervals u_k are of the same size, but this is not a compulsory restriction.

5.4.2 Feature selection

For the same set of $|\mathcal{U}|$ time intervals, we retrieve – from authoritative sources – the corresponding values of the target variable $y^{(\mathcal{U})}$:

$$y^{(\mathcal{U})} = [y_1 \text{ ... } y_{|\mathcal{U}|}]^T. \tag{5.5}$$

$\mathcal{X}^{(\mathcal{U})}$ and $y^{(\mathcal{U})}$ are used as an input in Bolasso, the bootstrapped version of LASSO [4]. In each bootstrap, LASSO selects a subset of the candidates and at the end Bolasso, by intersecting the bootstrap outcomes, attempts to make this selection consistent. LASSO is formulated as follows:

$$\begin{aligned} \min_{w} \quad & \|\mathcal{X}^{(\mathcal{U})}w - y^{(\mathcal{U})}\|_2^2 \\ \text{s.t.} \quad & \|w\|_1 \leq t, \end{aligned} \tag{5.6}$$

where t is the regularisation parameter controlling the shrinkage of w's L1-norm. In turn, t can be expressed as

$$t = \alpha \cdot \|w_{\text{OLS}}\|_1, \;\; \alpha \in (0, 1], \tag{5.7}$$

where w_{OLS} is the OLS regression solution and α denotes the desired shrinkage percentage of w_{OLS}'s L1-norm. Bolasso's implementation applies LARS which is able to explore the entire regularisation path at the cost of one matrix inversion and decides the optimal value of the regularisation parameter (t or α) by either using the largest consistent region, *i.e.* the largest continuous range on the regularisation path, where the set of selected variables remains the same, or by cross-validating [39]. We explain Bolasso in detail in Section 5.4.4, when looking at the entire algorithm (Algorithm 4) that encapsulates our method.

5.4.3 Inference and consensus threshold

After selecting a subset $\mathcal{F} = \{f_i\}$, $i \in \{1, ..., |\mathcal{F}|\}$ of the feature space $\mathcal{C}$ ($\mathcal{F} \subseteq \mathcal{C}$), the VSR of the initial vocabulary $\mathcal{X}^{(\mathcal{U})}$ is reduced to an array $\mathcal{Z}^{(\mathcal{U})}$ of size $|\mathcal{U}| \times |\mathcal{F}|$. We learn the weights of the selected features by performing OLS regression:

$$\min_{w_s} \quad \|(\mathcal{Z}^{(\mathcal{U})} w_s + \beta) - y^{(\mathcal{U})}\|_2^2, \tag{5.8}$$

where vector w_s denotes the learnt weights for the selected features and scalar β is regression's bias term.

A strict application of Bolasso implies that only features with a non zero weight in all bootstraps are going to be considered. In our methodology a soft version of Bolasso is applied (named as *Bolasso-S* in [4]), where features are considered if they acquire a non zero weight in a fraction of the bootstraps, which is referred to as Consensus Threshold (**CT**). CT ranges in $(0, 1]$ and obviously is equal to 1 in the strict application of Bolasso. The value of CT, expressed by a percentage, is decided using a validation set. To constrain the computational complexity of the learning phase, we consider 21 discrete CTs from 50% to 100% with a step of 2.5%.

Overall, the experimental process includes three steps: (**a**) training, where for each CT we retrieve a set of selected features from Bolasso and their weights from OLS regression, (**b**) validating CT, where we select the optimal CT value based on a validation set, and (**c**) testing, where the performance of our previous choices is computed. Training, validation and testing sets are by definition disjoint from each other.

The MSE between inferred ($\mathcal{X}w$) and target values (y) defines the loss ($\mathcal{L}$) during all steps. For a sample of size $|\mathcal{D}|$ this is defined as:

$$\mathcal{L}(w) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \ell(\langle x_i w \rangle, y_i), \tag{5.9}$$

where the loss function $\ell(\langle x_i w \rangle, y_i) = (\langle x_i w \rangle - y_i)^2$. This decision comes naturally from the definition of LASSO, which essentially tries to find the w's that minimise the squared loss plus a regularisation factor. Although the squared loss tends to penalise outliers more than other loss functions (*e.g.* the absolute loss), it is the one that has been investigated thoroughly in the relevant literature and research.[2] The Root Mean Square Error (**RMSE**) has been used as a more comprehensive metric (it has the same units with the target variables) for presenting results in the next sections.

To summarise CT's validation, suppose that for all considered consensus thresholds CT$_i$, $i \in \{1, ..., 21\}$, training yields $\mathcal{F}_i$ sets of selected features respectively, whose losses on the validation set are denoted by $\mathcal{L}_i^{(\text{val})}$. Then, if $i^{(*)}$ denotes the index of the selected CT and set of features, then it is given by:

$$i^{(*)} = \arg\min_i \mathcal{L}_i^{(\text{val})}. \tag{5.10}$$

Therefore, CT$_{i^{(*)}}$ is the result of the validation process and $\mathcal{F}_{i^{(*)}}$ is used in the testing phase.

Taking into consideration that both target values (rainfall and flu rates) can only be zero or positive, we threshold the negative inferred values with zero during testing, *i.e.* we set $x_i \leftarrow \max\{x_i, 0\}$. We perform this filtering only in the testing phase; during CT's validation, we want to keep track of deviations in the negative space as well.

5.4.4 Soft-Bolasso with consensus threshold validation

In this section we go through Algorithm 4, which describes the core of our approach. The algorithm takes as input the set of candidate features, training and validation sets with observations and ground truth, a set of consensus thresholds (CT) that are going to be explored, the number of bootstraps, an optional numeric parameter *maxFeaturesNum* that causes an

[2] For example the LARS algorithm that has been applied in our work uses the squared loss.

early stop in every bootstrap of LASSO (when this number of features is reached, LASSO terminates), and another numeric parameter *percentLCR*, the use of which is explained later on. The observations are assumed to be standardised and the responses centred. The outputs of the algorithm are the set of selected candidate features and the optimal consensus threshold.

First, a set of λ's is formed by performing LARS on the entire training set as a basis and then by re-applying LARS on random samples of the training set. In every bootstrap, the training set is sampled uniformly with replacement and then used as input to LARS. After executing all bootstraps, we end up with a set of weights for each bootstrap and value of λ. By identifying the nonzero weights over all bootstraps, we get a set of selected features for each pair of λ_i and CT$_i$.

The next step is to decide the optimal value of λ for each CT$_i$. We use two methods for this task: either the largest consistent region (LCR) on the regularisation path where the selected values do not change, or if less than *percentLCR*% of the specified λ's lie in the LCR, 5-fold cross-validation. In our experiments, we set *percentLCR* to 0.05 (5%). LCR is in general a preferable choice as the creation of a consistent region for the selected features is the main aim of Bolasso. Hence, if such a region exists, there is evidence that Bolasso's solution has consistent characteristics and therefore it shall be used [4].

Finally, after computing the optimal value of λ for each CT$_i$, and therefore retrieving the corresponding optimal set of selected features, a validation set is used for deciding the optimal CT, and therefore the set of selected features.

5.4.5 Extracting candidate features and forming feature classes

Three classes of candidate features have been investigated: unigrams or 1-grams (denoted by ***U***), bigrams or 2-grams (***B***) and a hybrid combination (***H***) of 1-grams and 2-grams. 1-grams being single words cannot be characterised by a consistent semantic interpretation in most of the topics. They take different meanings and express distinct outcomes based on the surrounding textual context. 2-grams on the other hand can be more focused semantically. However, their frequency in a corpus is expected to be lower than the one of 1-grams. Particularly, in the Twitter corpus which consists of very short pieces of text (tweets are at most 140 characters long), the daily frequency for some of them is sometimes close to zero.

ALGORITHM 4: Soft-Bolasso with Consensus Threshold Validation

```
Input: $\mathcal{C}_{1:n}$, $\mathcal{X}^{(train)}_{[1:m,1:n]}$, $y^{(train)}_{1:m}$, $\mathcal{X}^{(val)}_{[1:m',1:n]}$, $y^{(val)}_{1:m'}$, $\mathrm{CT}_{1:v}$, $B$, maxFeaturesNum, percentLCR
Output: $\hat{\mathcal{C}}_{1:p}$, $\hat{\mathrm{CT}}$
$\lambda_{1:k} \leftarrow$ computeLambdas $\left(\mathcal{X}^{(train)}_{[1:m,1:n]}, y^{(train)}_{1:m}\right)$;
for $i \leftarrow 1$ to $B$ do
    $\mathcal{X}^{(samp_i)}_{[1:m,1:n]}, y^{(samp_i)}_{1:m} \leftarrow$ sample $\left(\mathcal{X}^{(train)}_{[1:m,1:n]}, y^{(train)}_{1:m}\right)$;
    $\mathcal{W}^{(i)}_{[1:k,1:n]} \leftarrow$ LARS $\left(\mathcal{X}^{(samp_i)}_{[1:m,1:n]}, y^{(samp_i)}_{1:m}, \lambda_{1:k}, \text{maxFeaturesNum}\right)$;
end
for $i \leftarrow 1$ to $k$ do                                              // $\forall\ \lambda_i$
    for $j \leftarrow 1$ to $v$ do                                          // $\forall$ CT$_j$
        $\dot{\mathcal{C}}^{(i,j)}_{1:z_{i,j}} \leftarrow$ getFeatures $\left(\mathcal{W}^{(1:B)}_{[i,1:n]}, \mathrm{CT}_j\right)$;      // $|z_{i,j}|$ varies per iteration
    end
end
// Optimal value for the regularisation parameter
for $j \leftarrow 1$ to $v$ do
    $\ddot{\mathcal{C}}^{(j)}_{1:s_j}, \lambda_{\mathrm{LCR}_j} \leftarrow$ LCR $\left(\lambda_{1:k}, \dot{\mathcal{C}}^{(1:k,j)}_{1:z_{1:k,j}}\right)$;         // $|s_j|$ varies per iteration
    if $\left(|\lambda_{\mathrm{LCR}_j}| < k * \text{percentLCR}\right)$ then
        $\ddot{\mathcal{C}}^{(j)}_{1:s_j} \leftarrow$ CrossVal $\left(\dot{\mathcal{C}}^{(1:k,j)}_{1:z_{1:k,j}}, \mathcal{X}^{(train)}_{[1:m,1:n]}, y^{(train)}_{1:m}\right)$;
    end
end
// Optimal value for the consensus threshold
$\hat{\mathcal{C}}_{1:p}, \hat{\mathrm{CT}} \leftarrow$ validate $\left(\ddot{\mathcal{C}}^{(1:v)}_{1:s_{1:v}}, \mathcal{X}^{(val)}_{[1:m',1:n]}, y^{(val)}_{1:m'}\right)$;
return $\hat{\mathcal{C}}_{1:p}$, $\hat{\mathrm{CT}}$;
```

The hybrid class of features exploits the advantages of classes U and B and reduces the impact of their disadvantages. It is formed by combining the training results of U and B for all CTs. We have tried two approaches for forming a hybrid set of features, described in Algorithms 5 and 6, and denoted by H and H_{II} respectively.

Suppose that for all considered consensus thresholds CT$_i$, $i \in \{1, ..., |\text{CT}|\}$, 1-grams and 2-grams selected via Bolasso are denoted by $\mathcal{F}_i^{(U)}$ and $\mathcal{F}_i^{(B)}$ respectively. Then, in the first hybrid approach (H) the pseudo-selected n-grams for all CTs for the hybrid class $\mathcal{F}_i^{(H)}$ are

ALGORITHM 5: Hybrid Combination of 1-grams and 2-grams (H)

```
Input: $\hat{\mathcal{C}}^{(1\text{g})}_{[1:v,1:p_{1:v}]}$, $\hat{\mathcal{C}}^{(2\text{g})}_{[1:v,1:q_{1:v}]}$, $\mathcal{X}^{(train,1\text{g})}_{[1:m,1:n_{1\text{g}}]}$, $\mathcal{X}^{(train,2\text{g})}_{[1:m,1:n_{2\text{g}}]}$, $y^{(train)}_{1:m}$, $\mathcal{X}^{(val,1\text{g})}_{[1:m',1:n_{1\text{g}}]}$, $\mathcal{X}^{(val,2\text{g})}_{[1:m',1:n_{2\text{g}}]}$,
       $y^{(val)}_{1:m'}$, $\text{CT}_{1:v}$
Output: $\hat{\text{CT}}^{(\text{h})}$, $\hat{\mathcal{C}}_{1:p_{\text{minL}}+q_{\text{minL}}}$
for $i \leftarrow 1$ to $v$ do
    $\mathcal{X}^{(train_i)}_{[1:m,1:p_i+q_i]} \leftarrow \mathcal{X}^{(train,1\text{g})}_{[1:m,\hat{\mathcal{C}}^{(1\text{g})}_{[i,1:p_i]}]} \cup \mathcal{X}^{(train,2\text{g})}_{[1:m,\hat{\mathcal{C}}^{(2\text{g})}_{[i,1:q_i]}]}$;
    $\mathcal{X}^{(val_i)}_{[1:m,1:p_i+q_i]} \leftarrow \mathcal{X}^{(val,1\text{g})}_{[1:m,\hat{\mathcal{C}}^{(1\text{g})}_{[i,1:p_i]}]} \cup \mathcal{X}^{(val,2\text{g})}_{[1:m,\hat{\mathcal{C}}^{(2\text{g})}_{[i,1:q_i]}]}$;
    $\mathcal{W}_{[i,1:p_i+q_i]} \leftarrow \text{OLS}\left(\mathcal{X}^{(train_i)}_{[1:m,1:p_i+q_i]}, y^{(train)}_{1:m}\right)$;
    $\mathcal{L}_i = \text{MSE}\left(\mathcal{X}^{(val_i)}_{[1:m',1:p_i+q_i]}\mathcal{W}_{[i,1:p_i+q_i]}, y^{(val)}_{1:m'}\right)$;
end
// Retrieve CT's index where the minimum loss occurs
$\text{minL} \leftarrow \arg\min_i \mathcal{L}_{1:v}$;
$\hat{\text{CT}}^{(\text{h})} \leftarrow \text{CT}_{\text{minL}}$;
$\hat{\mathcal{C}}_{1:p_{\text{minL}}+q_{\text{minL}}} \leftarrow \hat{\mathcal{C}}^{(1\text{g})}_{[\text{minL},1:p_{\text{minL}}]} \cup \hat{\mathcal{C}}^{(2\text{g})}_{[\text{minL},1:q_{\text{minL}}]}$;
return $\hat{\text{CT}}^{(\text{h})}$, $\hat{\mathcal{C}}_{1:p_{\text{minL}}+q_{\text{minL}}}$;
```

formed by their one-by-one union, $\mathcal{F}_i^{(H)} = \{\mathcal{F}_i^{(U)} \cup \mathcal{F}_i^{(B)}\}$, $i \in \{1, ..., |\text{CT}|\}$. Likewise, $\mathcal{Z}_i^{(H)} = \{\mathcal{Z}_i^{(U)} \cup \mathcal{Z}_i^{(B)}\}$, $i \in \{1, ..., |\text{CT}|\}$, where $\mathcal{Z}$ denotes the VSR of each feature class (using the notation of Section 5.4.3). Validation and testing are performed on $\mathcal{Z}_i^{(H)}$ as it has already been described in Section 5.4.3.

In the second hybrid approach, the pseudo-selected n-grams are formed by exploring all possible $|\text{CT}|^2$ pairs of selected 1-grams and 2-grams. Thus, in that case $\mathcal{F}_i^{(H_{\text{II}})} = \{\mathcal{F}_i^{(U)} \cup \mathcal{F}_j^{(B)}\}$, $\forall i, j \in \{1, ..., |\text{CT}|\}$, and then this hybrid combination method proceeds in the same manner as the previous one. Note that compiling an optimal hybrid scheme is not the main focus here; our aim is to investigate whether a simple combination of 1-grams and 2-grams is able to deliver better results. The experimental results in the following sections do indeed indicate that feature class H performs on average better than U and B, something that does not always hold for H_{II}.

Finally, we create one more class of features by combining again all 1-grams and 2-grams, but this time we perform the entire training process – *i.e.* run the soft Bolasso algorithm – on

ALGORITHM 6: Hybrid Combination of 1-grams and 2-grams (H_{II})

```
Input: $\hat{\mathcal{C}}^{(1\text{g})}_{[1:v,1:p_{1:v}]}$, $\hat{\mathcal{C}}^{(2\text{g})}_{[1:v,1:q_{1:v}]}$, $\mathcal{X}^{(train,1\text{g})}_{[1:m,1:n_{1\text{g}}]}$, $\mathcal{X}^{(train,2\text{g})}_{[1:m,1:n_{2\text{g}}]}$, $y^{(train)}_{1:m}$, $\mathcal{X}^{(val,1\text{g})}_{[1:m',1:n_{1\text{g}}]}$, $\mathcal{X}^{(val,2\text{g})}_{[1:m',1:n_{2\text{g}}]}$,
       $y^{(val)}_{1:m'}$, $\text{CT}_{1:v}$
Output: $\hat{\text{CT}}^{(\text{h},1\text{g})}$, $\hat{\text{CT}}^{(\text{h},2\text{g})}$, $\hat{\mathcal{C}}_{1:p_{\text{minL\_1g}}+q_{\text{minL\_2g}}}$
for $i \leftarrow 1$ to $v$ do
    for $j \leftarrow 1$ to $v$ do
        $\mathcal{X}^{(train_{i,j})}_{[1:m,1:p_i+q_j]} \leftarrow \mathcal{X}^{(train,1\text{g})}_{[1:m,\hat{\mathcal{C}}^{(1\text{g})}_{[i,1:p_i]}]} \cup \mathcal{X}^{(train,2\text{g})}_{[1:m,\hat{\mathcal{C}}^{(2\text{g})}_{[j,1:q_j]}]}$;
        $\mathcal{X}^{(val_{i,j})}_{[1:m,1:p_i+q_j]} \leftarrow \mathcal{X}^{(val,1\text{g})}_{[1:m,\hat{\mathcal{C}}^{(1\text{g})}_{[i,1:p_i]}]} \cup \mathcal{X}^{(val,2\text{g})}_{[1:m,\hat{\mathcal{C}}^{(2\text{g})}_{[j,1:q_j]}]}$;
        $\mathcal{W}^{(i,j)}_{[1:p_i+q_j]} \leftarrow \text{OLS}\left(\mathcal{X}^{(train_{i,j})}_{[1:m,1:p_i+q_j]}, y^{(train)}_{1:m}\right)$;
        $\mathcal{L}_{i,j} = \text{MSE}\left(\mathcal{X}^{(val_{i,j})}_{[1:m',1:p_i+q_j]}\mathcal{W}^{(i,j)}_{[1:p_i+q_j]}, y^{(val)}_{1:m'}\right)$;
    end
end
// Retrieve CT's index where the minimum loss occurs
$\text{minL\_1g}, \text{minL\_2g} \leftarrow \arg\min_{i,j} \mathcal{L}_{[1:v,1:v]}$;
$\hat{\text{CT}}^{(\text{h},1\text{g})} \leftarrow \text{CT}_{\text{minL\_1g}}$;
$\hat{\text{CT}}^{(\text{h},2\text{g})} \leftarrow \text{CT}_{\text{minL\_2g}}$;
$\hat{\mathcal{C}}_{1:p_{\text{minL\_1g}}+q_{\text{minL\_2g}}} \leftarrow \hat{\mathcal{C}}^{(1\text{g})}_{[\text{minL\_1g},1:p_{\text{minL\_1g}}]} \cup \hat{\mathcal{C}}^{(2\text{g})}_{[\text{minL\_2g},1:q_{\text{minL\_2g}}]}$;
return $\hat{\text{CT}}^{(\text{h},1\text{g})}$, $\hat{\text{CT}}^{(\text{h},2\text{g})}$, $\hat{\mathcal{C}}_{1:p_{\text{minL\_1g}}+q_{\text{minL\_2g}}}$;
```

this unified data set. This method – conventionally named as UB – is expected to perform worse than the hybrid combinations due to the fact that it increases the dimensionality of the problem at hand without providing more training points. Indeed, as we will see in the next sections, the first hybrid approach (H) performs better than all the other class features.

5.5 Comparing with other methodologies

As part of the evaluation process, we compare our results with a baseline approach which encapsulates the methodologies in [56] and [31]. Those approaches, as explained in Section 5.3, mainly differ in the feature selection process which is performed via correlation analysis; Algorithm 7 describes this process in detail.

ALGORITHM 7: Baseline Method: Feature Selection via Correlation Analysis

```
Input: $\mathcal{C}_{1:n}$, $\mathcal{X}^{(train)}_{[1:m,1:n]}$, $y^{(train)}_{1:m}$, $\mathcal{X}^{(val)}_{[1:m',1:n]}$, $y^{(val)}_{1:m'}$
Output: $\hat{\mathcal{C}}_{1:p}$
$\rho_{1:n} \leftarrow$ correlation $\left(\mathcal{X}^{(train)}_{[1:m,1:n]}, y^{(train)}_{1:m}\right)$;
$\hat{\rho}_{1:n} \leftarrow$ descendingCorrelationIndex$(\rho_{1:n})$;
$\hat{\mathcal{C}}_{1:n} \leftarrow \mathcal{C}_{\hat{\rho}_{1:n}}$;
$i \leftarrow 1$;
while $i \leq k$ do
    $\mathcal{L}_i \leftarrow$ validate $\left(\mathcal{X}^{(train)}_{[1:m,\hat{\rho}_{1:i}]}, y^{(train)}_{1:m}, \mathcal{X}^{(val)}_{[1:m',\hat{\rho}_{1:i}]}, y^{(val)}_{1:m'}\right)$;
    $i \leftarrow i + 1$;
end
$p \leftarrow \arg\min_{i} \mathcal{L}_i$;
return $\hat{\mathcal{C}}_{1:p}$;
```

Briefly, given a set $\mathcal{C}$ of n candidate features, their computed VSRs for training and validation $\mathcal{X}^{(train)}$ and $\mathcal{X}^{(val)}$ and the corresponding response values $y^{(train)}$ and $y^{(val)}$, this feature selection process: **a)** computes the Pearson correlation coefficients (ρ) between each candidate feature and the response values in the training set, **b)** ranks the retrieved correlation coefficients in descending order, **c)** computes the OLS-fit loss ($\mathcal{L}$) of incremental subsets of the top-k correlated terms on the validation set and **d)** selects the subset of candidate features with the minimum loss. The inference performance of the selected features is evaluated on a (disjoint) test set.

5.6 Case study I: nowcasting rainfall rates from Twitter

In the first case study, we exploit the content of Twitter to infer daily rainfall rates (measured in millimetres of precipitation) for five UK cities, namely *Bristol*, *London*, *Middlesbrough*, *Reading* and *Stoke-on-Trent*. The choice of those locations has been based on the availability of ground truth, *i.e.* daily rainfall measurements from weather stations installed in their vicinity.

We consider the inference of precipitation levels at a given time and place as a good benchmark problem, in that it has many of the properties of other more useful scenarios, while still allows us to verify the performance of the system since rainfall is a measurable variable. The event of rain is a piece of information available to the significant majority of Twitter users and affects various activities that could form a discussion topic in tweets. Furthermore, predictions about it are not always easy due to its non smooth and sometimes unpredictable behaviour [78].

The candidate markers for this case study are extracted from weather related web references, such as Wikipedia's page on Rainfall, an English language course on weather vocabulary, a page with formal weather terminology and several others. The majority of the extracted candidate features is not directly related to the target topic, but there exists a subset of markers which could probably offer a good semantic interpretation. Markers with a count ≤ 10 in the Twitter corpus used for this case study are removed. Hence, from the extracted 2381 1-grams, 2159 have been kept as candidates; likewise the 7757 extracted 2-grams have been reduced to 930.[3]

5.6.1 Experimental settings

A year of Twitter data and rainfall observations (from the 1^{st} of July, 2009 to the 30^{th} of June, 2010) formed the input data for this experiment. For this time period and the considered locations, 8.5 million tweets have been collected. In each run of Bolasso the number of bootstraps is proportional to the size of the training sample (approximately 13% using the same principle as in [4]), and in every bootstrap we select at most 300 features by performing at most 900 iterations. A bootstrap is completed as soon as one of those two stopping criteria is met. This is an essential trade-off which guarantees a quicker execution of the learning phase, especially when dealing with large amounts of data.

The performance of each feature class (U, B, H, H_{II}, UB) is computed by applying a 6-fold cross validation. Each fold is based on 2 months of data starting from the month pair July-August (2009) and ending with May-June (2010). In every step of the cross validation, 5 folds are used for training, the first half (a month-long data) of the remaining fold for validating CT and the second half for testing the performance of the selected markers and

[3] Complete lists of the web references used and the extracted markers for the rainfall case study are available at http://geopatterns.enm.bris.ac.uk/twitter/rainfall-cf.php.

Table 5.1: Nowcasting Rainfall Rates – Derived Consensus Thresholds and numbers of selected features (in parentheses) for all Feature Classes (FC) in the rounds of 6-fold cross validation – Fold i denotes the validation/testing fold of round $7 - i$

FC	**Fold 6**	**Fold 5**	**Fold 4**	**Fold 3**	**Fold 2**	**Fold 1**
U	100% (*4*)	92.5% (*19*)	90% (*17*)	92.5% (*12*)	90% (*17*)	75% (*28*)
B	90% (*21*)	67.5% (*10*)	95% (*10*)	67.5% (*15*)	90% (*9*)	62.5% (*38*)
H	100% (*8*)	92.5% (*27*)	95% (*21*)	92.5% (*27*)	90% (*26*)	52.5% (*131*)
H_{II}	100/90% (*25*)	92.5/67.5% (*29*)	90/95% (*27*)	92.5/57.5% (*24*)	90/90% (*26*)	82.5/100% (*23*)
UB	100% (*5*)	65% (*29*)	95% (*8*)	75% (*25*)	87.5% (*12*)	75% (*40*)

their weights. Training is performed by using the VSRs of all five locations in a batch data set, CT's validation is carried out on the same principle (we learn the same markers-weights under the same CT for all locations), and finally testing is done both on the batch data set (to retrieve a total performance evaluation) as well as on each location separately. Finally, we also compute the inference performance of the baseline approach for feature selection (Algorithm 7) for the same training, validation and testing sets, considering the top $k = 300$ correlated terms.

5.6.2 Results

The derived CTs as well as the numbers of selected features for all rounds of the 6-fold cross validation are presented on Table 5.1. In most rounds CT values are high and as a result the number of selected features is relatively small. In turn, this could mean that in most occasions a few markers were able to capture the rainfall rates signal. The last round, where the validation data set is based on July 2009, is an exception, where we derive lower CTs for most feature classes as well as larger numbers of selected features. This can be interpreted by the fact that July is not only the 2nd most rainy month in our data set, but also a summer month – hence, tweets for rain could be followed or preceded by tweets discussing a sunny day, creating instabilities during the validation process. In addition, our data set is restricted to only one year of weather observations and therefore seasonal patterns like this one are not expected to be entirely captured.

Detailed performance evaluation results (total and per location for all feature classes) are presented on Table 5.2. For a better interpretation of the numerical values (in mm), consider that the average rainfall rate in our data set is equal to 1.8 with a standard deviation of 3.9 and a range of [0,65]. Our method outperforms the baseline approach (see Algorithm 7) in all-but-one intermediate RMSE indications as well as in total for all feature classes,

Table 5.2: Nowcasting Rainfall Rates – RMSEs (in *mm*) for all Feature Classes (FC) and locations in the rounds of 6-fold cross validation – Fold i denotes the validation/testing fold of round $7 - i$. The last column holds the RMSEs of the baseline method.

Location	**FC**	**Fold 6**	**Fold 5**	**Fold 4**	**Fold 3**	**Fold 2**	**Fold 1**	**Mean RMSE**	**BS-Mean RMSE**
Bristol	U	1.164	1.723	1.836	2.911	1.607	2.348	1.931	2.173
	B	1.309	1.586	2.313	3.371	1.59	1.409	1.93	2.218
	H	1.038	1.631	2.334	2.918	1.579	2.068	1.928	2.094
London	U	1.638	1.507	5.079	2.582	1.62	6.261	3.115	3.297
	B	1.508	5.787	4.887	3.403	1.478	6.568	3.939	4.305
	H	1.471	1.526	4.946	2.813	1.399	6.13	3.047	4.101
Middlesbrough	U	4.665	1.319	3.102	2.618	2.949	2.536	2.865	2.951
	B	4.355	1.069	3.379	2.22	2.918	2.793	2.789	2.946
	H	4.47	1.098	3.016	2.504	2.785	2.353	2.704	6.193
Reading	U	2.075	1.566	2.087	2.393	1.981	2.066	2.028	2.168
	B	0.748	2.74	1.443	3.016	1.572	3.429	2.158	2.159
	H	1.636	1.606	1.368	2.571	1.695	2.145	1.836	2.214
Stoke-on-Trent	U	3.46	1.932	1.744	4.375	2.977	1.962	2.742	2.855
	B	3.762	1.493	1.433	2.977	2.447	2.668	2.463	2.443
	H	3.564	1.37	1.499	3.785	2.815	1.931	2.494	2.564
Total RMSE	U	2.901	1.623	3.04	3.062	2.31	3.443	**2.73**	2.915
	B	2.745	3.062	2.993	3.028	2.083	3.789	**2.95**	3.096
	H	2.779	1.459	2.937	2.954	2.145	3.338	**2.602**	4.395
	H_{II}	2.68	3.009	2.851	2.962	2.145	3.461	**2.851**	–
	UB	2.875	3.157	3.006	2.972	2.071	3.553	**2.939**	–

achieving an improvement of 10.74% (derived by comparing the lowest total RMSEs for each method). The overall performance for our method indicates that feature class H performs better than all others, and interestingly unigrams outperform bigrams as well as the remaining hybrid feature classes.

Presenting all intermediate results for each round of the cross validation would have been intractable. In the remaining part of this Section, we present the results of learning and testing for cross-validation's round 5 only, where the month of testing is October, 2009. Tables 5.3, 5.4 and 5.5 list the selected features in alphabetical order together with their

Table 5.3: Feature Class U – 1-grams selected by Bolasso for Rainfall case study (Round 5 of 6-fold cross validation) – All weights (**w**) should be multiplied by 10^3

1-gram	**w**	**1-gram**	**w**	**1-gram**	**w**	**1-gram**	**w**	**1-gram**	**w**
flood	0.767	piss	0.247	rainbow	-0.336	todai	-0.055	wet	0.781
influenc	0.579	pour	1.109	sleet	1.766	town	-0.134		
look	-0.071	***puddl***	**3.152**	suburb	1.313	umbrella	0.223		
monsoon	2.45	rain	0.19	sunni	-0.193	wed	-0.14		

Table 5.4: Feature Class B – 2-grams selected by Bolasso for Rainfall case study (Round 5 of 6-fold cross validation) – All weights (**w**) should be multiplied by 10^3

2-gram	**w**	**2-gram**	**w**	**2-gram**	**w**	**2-gram**	**w**	**2-gram**	**w**
air travel	-2.167	light rain	2.508	raini dai	2.046	stop rain	3.843	wind rain	5.698
horribl weather	3.295	***pour rain***	**7.161**	rain rain	4.490	sunni dai	-0.97		

Table 5.5: Feature Class H – Hybrid selection of 1-grams and 2-grams for Rainfall case study (Round 5 of 6-fold cross validation) – All weights (**w**) should be multiplied by 10^3

***n*-gram**	**w**	***n*-gram**	**w**	***n*-gram**	**w**	***n*-gram**	**w**	***n*-gram**	**w**
air travel	-1.841	monsoon	2.042	rain rain	2.272	sunni	-0.125	wet	0.524
flood	0.781	piss	0.24	rainbow	-0.294	sunni dai	-0.165	***wind rain***	**3.399**
horribl weather	1.282	pour	0.729	raini dai	1.083	todai	-0.041		
influenc	0.605	pour rain	2.708	sleet	1.891	town	-0.112		
light rain	2.258	puddl	3.275	stop rain	2.303	umbrella	0.229		
look	-0.067	rain	0.122	suburb	1.116	wed	-0.1		

weights for feature classes U, B and H respectively.[4] For class H we have also compiled a word cloud with the selected features as a more comprehensive representation of the selection outcome (Figure 5.2). The majority of the selected 1-grams (Table 5.3) has a very close semantic connection with the underlying topic – stem '*puddl*' holds the largest weight, whereas stem '*sunni*' has taken a negative weight and interestingly, the word '*rain*' has a relatively small weight. There also exist a few words without a direct semantic connection, but the majority of them has negative weights and in a way they might be acting as mitigators of non weather related uses of the remaining rainy weather oriented and positively weighted features. The selected 2-grams (Table 5.4) have a clearer semantic connection with the topic; '*pour rain*' acquires the highest weight. In this particular case, the features for class H are formed by the exact union of the ones in classes U and B, but take different weights (Table 5.5 and Figure 5.2).

[4] We do not show detailed results for the feature classes H_{II} and UB since feature class H – which is based on the same principle of combining 1-grams and 2-grams – outperforms them.

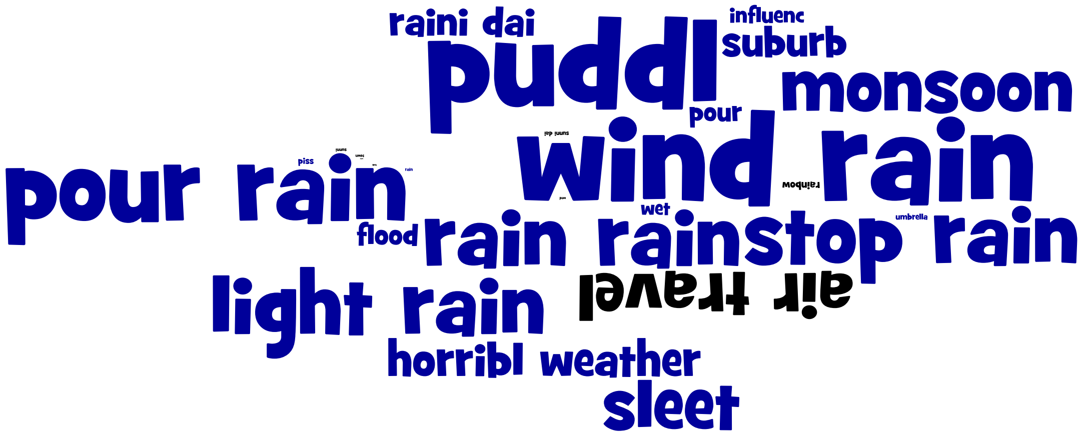
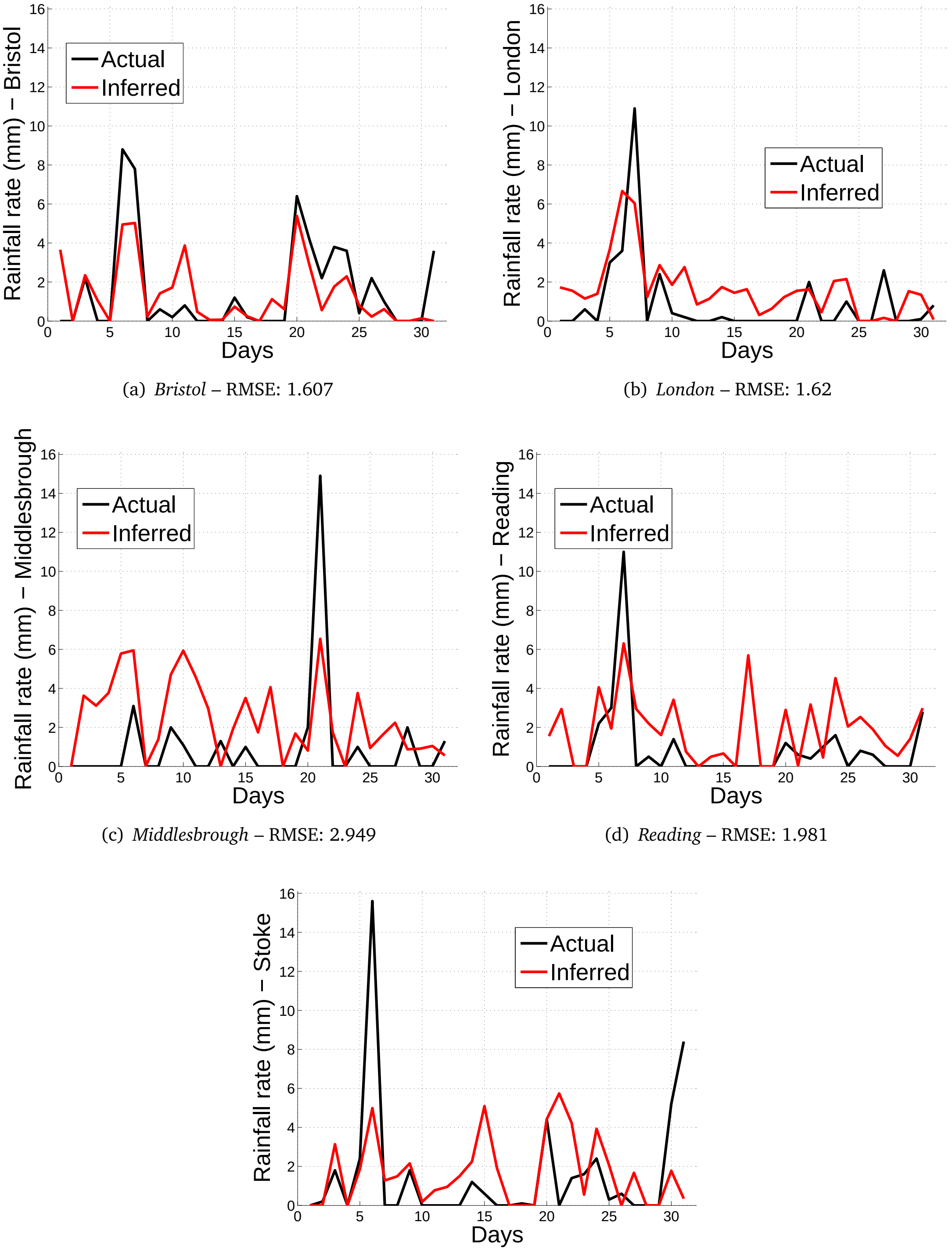


Figure 5.2: Table 5.5 in a word cloud, where font size is proportional to regression's weight and flipped words have negative weights

Inference results per location for cross validation's round 5 are presented in Figures 5.3, 5.4 and 5.5 for U, B, and H feature classes respectively. Overall, inferences follow the pattern of actual rain; for feature class B, we see that inferences in some occasions appear to have a positive lower bound (see Figure 5.4(e)), which is actually the positive bias term of OLS regression appearing when the selected markers have zero frequencies in the daily Twitter corpus of a location. As mentioned before, this problem is resolved in H since it is very unlikely for 1-grams to also have a zero frequency. Results for class H, depicted in Figures 5.5(a) (*Bristol*), 5.5(b) (*London*) and 5.5(d) (*Reading*), demonstrate a good fit with the target signal.

5.7 Case study II: nowcasting flu rates from Twitter

In the second case study, we use the content of Twitter to infer regional flu rates in the UK. We base our inferences in three UK regions, namely *Central England* & *Wales*, *North England* and *South England*. Ground truth, *i.e.* official flu rate measurements, is derived from HPA. HPA's weekly reports are based on information collected from the RCGP and express the number of GP consultations per 100,000 citizens, where the result of the diagnosis was ILI. According to HPA, a flu rate less than or equal to 30 is considered as baseline, flu rates below



(a) *Bristol* – RMSE: 1.607

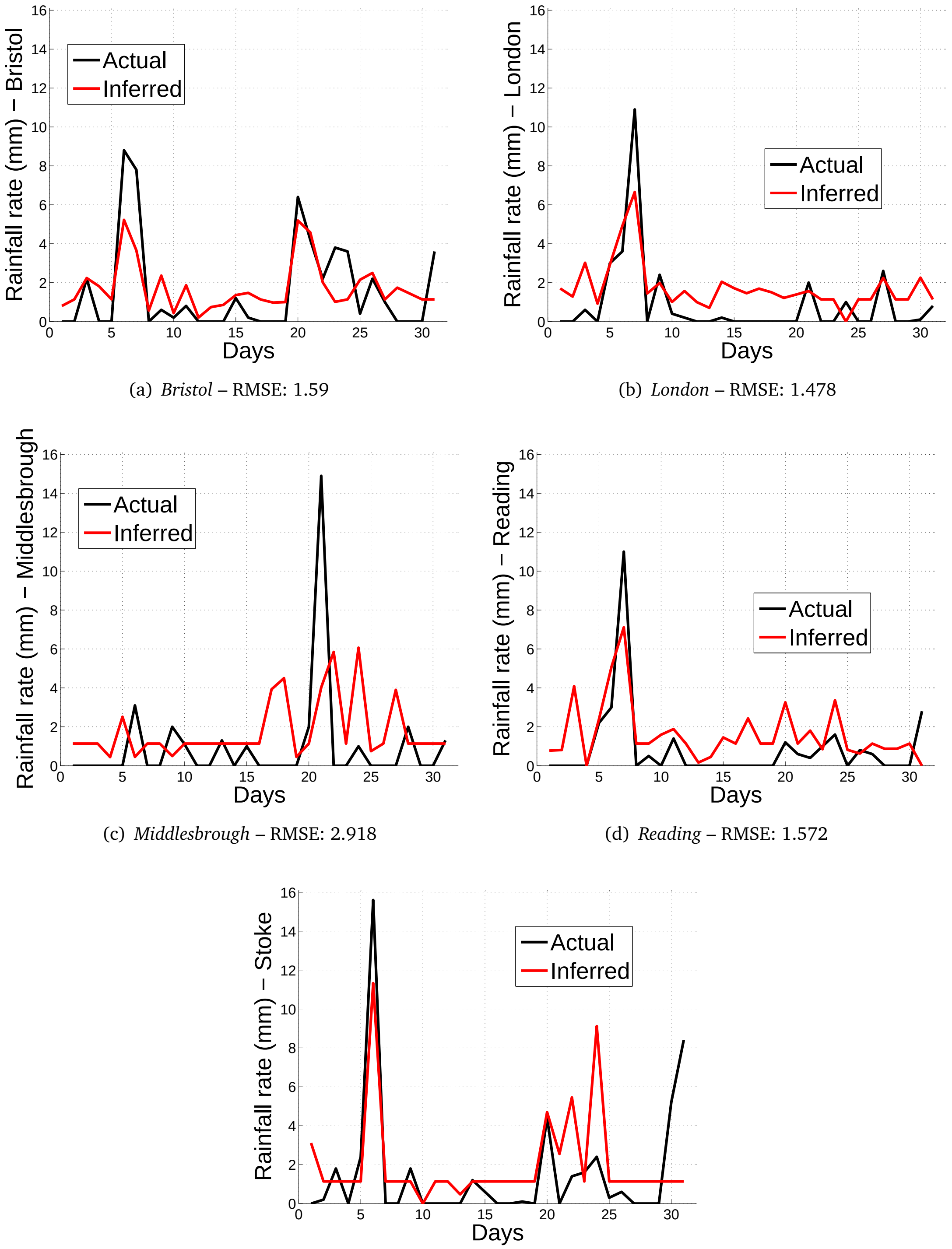
(b) *London* – RMSE: 1.62

(c) *Middlesbrough* – RMSE: 2.949

(d) *Reading* – RMSE: 1.981

(e) *Stoke-on-Trent* – RMSE: 2.977

Figure 5.3: Feature Class U – Inference for Rainfall case study (Round 5 of 6-fold cross validation)



(a) *Bristol* – RMSE: 1.59

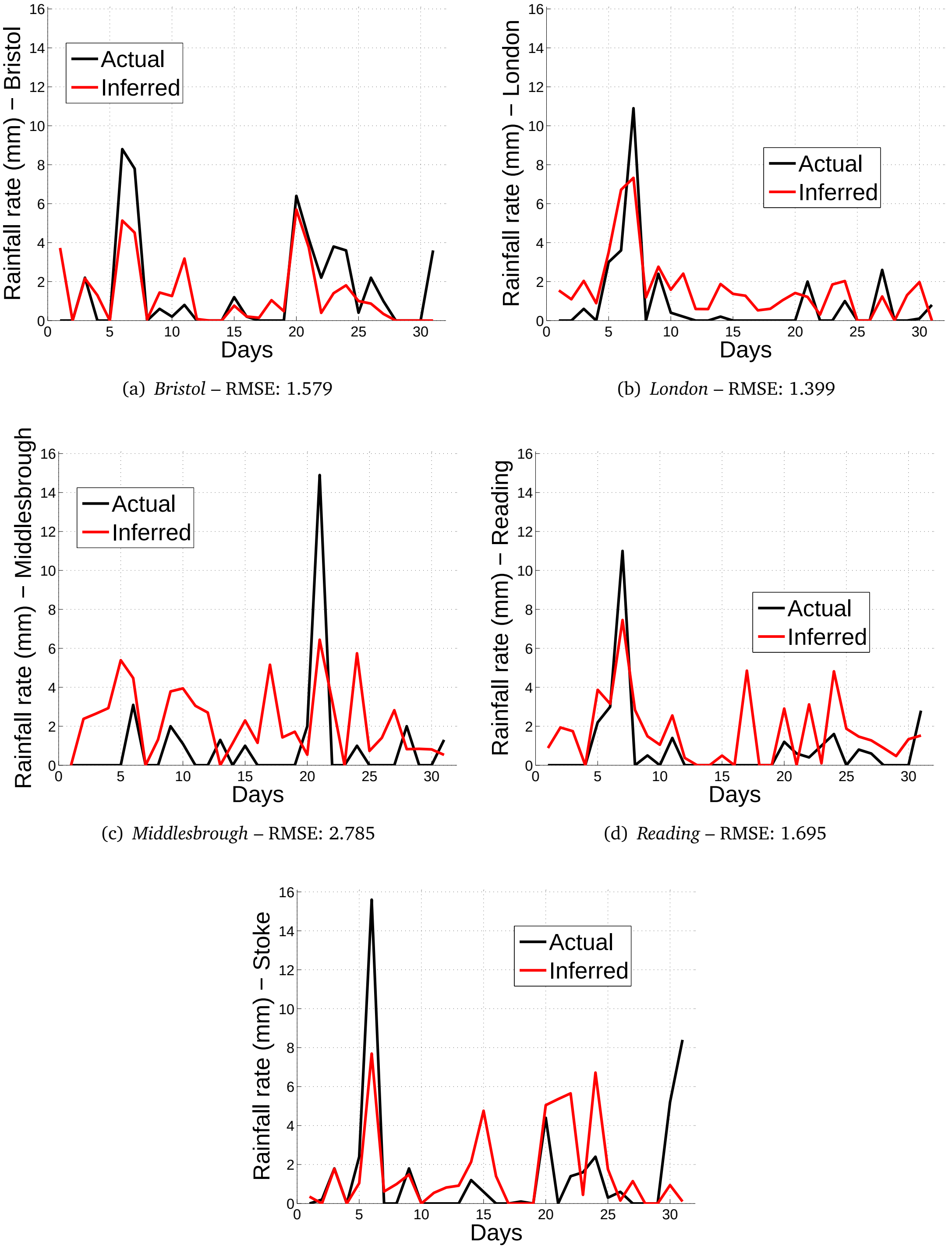
(b) *London* – RMSE: 1.478

(c) *Middlesbrough* – RMSE: 2.918

(d) *Reading* – RMSE: 1.572

(e) *Stoke-on-Trent* – RMSE: 2.447

Figure 5.4: Feature Class B – Inference for Rainfall case study (Round 5 of 6-fold cross validation)



(a) *Bristol* – RMSE: 1.579

(b) *London* – RMSE: 1.399

(c) *Middlesbrough* – RMSE: 2.785

(d) *Reading* – RMSE: 1.695

(e) *Stoke-on-Trent* – RMSE: 2.815

Figure 5.5: Feature Class H – Inference for Rainfall case study (Round 5 of 6-fold cross validation)

100 are normal, between 100 and 200 are above average and over 200 are characterised as exceptional.[5] To create a daily representation of HPA's weekly reports we use linear interpolation between the weekly rates. Given the flu rates r_i and r_{i+1} of two consecutive weeks, we compute a step factor δ from

$$\delta = \frac{r_{i+1} - r_i}{7}, \tag{5.11}$$

and then produce flu rates for the days in between using the equation

$$d_j = d_{j-1} + \delta,\, j \in \{2, ..., 7\}, \tag{5.12}$$

where d_j denotes the flu rate of week's day j and $d_1 = r_i$. The main assumption here is that a regional flu rate will be monotonically increasing or decreasing within the duration of a week.

Candidate features are extracted from several web references, such as the web sites of National Health Service, BBC, Wikipedia and so on, following the general principle, *i.e.* including encyclopedic, scientific and more informal input. Similarly to the previous case study, extracted 1-grams are reduced from 2428 to 2044, and extracted 2-grams from 7589 to 1678. Here, we have removed n-grams with a count ≤ 50 since the number of tweets involved is approximately 5 times larger compared to the rainfall case study.[6]

In the flu case study, not enough peaks are present in the ground truth time series, as the collected Twitter data cover only one flu period with above average rates (Swine Flu epidemic in June-July 2009). During performance evaluation, this results in training mostly on non-flu periods where there is no strong flu signal – hence, feature selection under those conditions is not optimal. To overcome this and assess properly the proposed methodology, we perform a random permutation of all data points based on their day index. The randomly permuted result for *South England*'s flu rate is shown on Figure 5.11(c) – we apply the same randomised index on all regions during performance evaluation.

[5] "Interpreting the HPA Weekly National Influenza Report", July 2009 – http://goo.gl/GWZmB.

[6] Complete lists of the web references and the extracted markers for the flu case study are available at http://geopatterns.enm.bris.ac.uk/twitter/influenza-cf.php.

Table 5.6: Nowcasting Flu Rates – Derived Consensus Thresholds and numbers of selected features (in parentheses) for all Feature Classes (FC) in the rounds of 5-fold cross validation – Fold i denotes the validation/testing fold of round $6 - i$

FC	Fold 5	Fold 4	Fold 3	Fold 2	Fold 1
U	52.5% (*90*)	52.5% (*100*)	52.5% (*108*)	62.5% (*67*)	50% (*62*)
B	55% (*42*)	62.5% (*47*)	92.5% (*14*)	85% (*10*)	52.5% (*36*)
H	55% (*124*)	62.5% (*131*)	52.5% (*151*)	60% (*103*)	50% (*100*)
H_{II}	82.5/55% (*82*)	57.5/62.5% (*119*)	72.5/92.5% (*50*)	67.5/60% (*105*)	50/80% (*81*)
UB	90% (*26*)	85% (*33*)	60% (*29*)	100% (*11*)	57.5% (*128*)

5.7.1 Experimental Settings

For this experiment, we considered tweets and ground truth in the time period between the 21^{st} of June, 2009 and 19^{th} of April, 2010 (303 days). The total number of tweets used reaches approximately 50 million. Similarly to the previous case study, we are applying 5-fold cross validation using data of 60 or 61 days per fold. In each round of the cross-validation, 4 folds are used for training. From the remaining fold, 30 days of data are used for validating CT and the rest for testing. Bolasso settings are identical to the ones used in the rainfall case.

Notice that in the following experiments data points are not contiguous in terms of time since they have been permuted randomly based on their day index (as explained in the previous section). However, we have included an example at the end of the next section, where contiguous (time-wise) training, validating and testing data points have been used.

5.7.2 Results

The derived CTs and numbers of selected features for all rounds of the 5-fold cross validation are presented on Table 5.6. In the flu case study, most CTs (especially in U and H feature classes) get a value closer to the lower bound (50%) after validation, and on average more features (compared to the rainfall case) are being selected. This is due to either the existence of only one significant flu period in the ground truth data or the general inadequacy of 1-grams to describe the underlying topic as effectively as in the previous case study.

Table 5.7 holds the performance results for all rounds of the cross validation. For a more comprehensive interpretation of the numerical values consider that the average ILI rate across the regions used in our experiments is equal to 26.659 with a standard deviation of

Table 5.7: Nowcasting Flu Rates – RMSEs for all Feature Classes (FC) and locations in the rounds of 5-fold cross validation – Fold i denotes the validation/testing fold of round $6 - i$. The last column holds the RMSEs of the baseline method.

Region	**FC**	**Fold 5**	**Fold 4**	**Fold 3**	**Fold 2**	**Fold 1**	**Mean RMSE**	**BS-Mean RMSE**
Central England & *Wales*	U	11.781	9.005	16.147	13.252	10.912	12.219	12.677
	B	11.901	12.2	21.977	12.426	14.615	14.624	15.665
	H	8.36	8.826	14.618	12.312	12.62	11.347	11.691
North England	U	9.757	6.708	9.092	13.117	8.489	9.432	10.511
	B	9.659	9.969	10.716	12.057	8.699	10.22	12.299
	H	9.782	7.112	6.65	13.694	7.607	8.969	9.752
South England	U	9.599	8.285	13.656	14.673	11.061	11.455	13.617
	B	13.536	9.209	16.188	14.279	8.531	12.348	12.977
	H	9.86	7.881	13.448	14.34	8.872	10.88	12.768
Total RMSE	U	10.426	8.056	13.29	13.699	10.222	**11.139**	12.438
	B	11.806	10.536	16.93	12.958	10.986	**12.643**	13.815
	H	9.359	7.971	12.094	13.475	9.93	**10.566**	11.617
	H_{II}	10.237	7.699	12.918	13.417	9.963	**10.847**	–
	UB	9.121	8.176	14.371	14.729	31.168	**15.513**	–

29.270 and ranges in [2,172]. Again, feature class U performs better than B, whereas H outperforms all others. However, in this case study, feature class H_{II} is the second best in terms of performance. Similarly to the previous case study, our method improves on the performance of the baseline approach by a factor of 9.05%.

For this case study, we present all intermediate results for cross validation's round 1. Tables 5.8, 5.9 and 5.10 show the selected features for U, B and H feature classes respectively. From the selected 1-grams (Table 5.8), stem '*irrig*'[7] has the largest weight. Many illness related markers have been selected such as '*cough*', '*health*', '*medic*', '*nurs*', '*throat*' and so on, but there exist also words with no clear semantic relation. Surprisingly, stem '*flu*' has not been selected as a feature in this round (and has only been selected in round 5). On the contrary, almost all selected 2-grams (Table 5.9) can be considered as flu related – '*confirm swine*' has the largest weight for both feature classes B and H (Table 5.10 and Figure 5.6).

[7] *Irrigation* describes the procedure of cleaning a wound or body organ by flushing or washing out with water or a medicated solution (WordNet).

Table 5.8: Feature Class U – 1-grams selected by Bolasso for Flu case study (Round 1 of 5-fold cross validation) – All weights ($\mathbf{w}$) should be multiplied by 10^4

1-gram	**w**	**1-gram**	**w**	**1-gram**	**w**	**1-gram**	**w**	**1-gram**	**w**
acut	-1.034	cleav	0.735	hippocr	-6.249	properti	-0.66	speed	-0.286
afford	-0.181	complex	-0.499	holidai	-0.017	psycholog	-1.103	spike	0.145
allergi	-2.569	cough	0.216	huge	-0.33	public	0.212	stage	0.109
approv	-0.672	cruis	-1.105	***irrig***	**10.116**	radar	0.284	strength	0.873
artifici	2.036	daughter	0.187	item	-0.337	reach	0.247	strong	0.336
assembl	0.589	dilut	4.165	knock	0.261	reliev	-0.254	swine	1.262
asthmat	4.526	drag	0.098	lethal	-0.73	remain	-0.755	tast	0.13
attempt	0.375	erad	0.201	major	-0.367	rough	0.068	team	-0.031
behavior	-1.747	face	-0.008	medic	1.06	run	0.242	throat	0.07
better	0.066	fellow	0.542	member	0.354	rush	-0.159	tissu	0.533
bind	0.675	fluid	2.002	mercuri	-0.588	scari	0.198	transmit	1.352
blood	0.059	fuss	0.575	metro	-0.397	seal	-0.161	troop	0.532
boni	1.308	germ	0.211	mile	-0.081	season	-0.103	typic	0.585
bulg	-0.966	guilti	-0.608	miss	0.071	seizur	2.448	underli	0.774
caution	2.578	habit	0.619	nurs	0.223	self	0.127	unquot	8.901
cellular	-2.125	halt	1.472	perform	0.084	sik	-0.634	upcom	0.642
checklist	-1.494	harbour	-0.472	personnel	-1.451	site	0.042	wave	0.042
chicken	0.317	health	-0.241	pictur	-0.134	soak	0.413	wikipedia	0.824

Table 5.9: Feature Class B – 2-grams selected by Bolasso for Flu case study (Round 1 of 5-fold cross validation) – All weights ($\mathbf{w}$) should be multiplied by 10^4

2-gram	**w**	**2-gram**	**w**	**2-gram**	**w**	**2-gram**	**w**
case swine	12.783	flu bad	6.641	need take	0.887	talk friend	-4.9
check code	6.27	flu jab	4.66	pain night	14.149	time knock	10.002
check site	0.568	flu relat	10.948	physic emotion	7.95	total cost	-11.582
confirm swine	**31.509**	flu symptom	7.693	sleep well	1.319	underli health	25.535
cough fit	7.381	flu web	-8.017	sore head	4.297	virus epidem	-28.204
cough lung	7.974	ground take	-15.208	spread viru	20.871	visit doctor	-12.327
cough night	16.73	health care	-0.636	stai indoor	5.482	weight loss	-0.447
die swine	9.722	healthcar worker	3.876	suspect swine	3.863	woke sweat	-33.133
effect swine	27.675	home wors	22.167	swine flu	1.153	wonder swine	11.5085
feel better	0.655	ion channel	9.755	symptom swine	5.895		
feel slightli	1.712	kick ass	-0.335	take care	0.382		

Table 5.10: Feature Class H – Hybrid selection of 1-grams and 2-grams for Flu case study (Round 1 of 5-fold cross validation) – All weights ($\mathbf{w}$) should be multiplied by 10^4

***n*-gram**	**w**	***n*-gram**	**w**	***n*-gram**	**w**	***n*-gram**	**w**
acut	-0.796	effect swine	19.835	medic	0.48	spike	0.032
afford	-0.106	erad	0.27	member	0.169	spread viru	12.918
allergi	-2.332	face	0.012	mercuri	-0.414	stage	0.101
approv	-0.516	feel better	0.15	metro	-0.365	stai indoor	1.969
artifici	1.319	feel slightli	0.775	mile	-0.092	strength	0.739
assembl	0.231	fellow	0.319	miss	0.073	strong	0.018
asthmat	2.607	flu bad	4.953	need take	0.759	suspect swine	2.503
attempt	0.322	flu jab	-0.11	nurs	0.118	swine	-0.203
behavior	-1.349	flu relat	3.183	pain night	9.823	swine flu	1.577
bind	0.437	flu symptom	1.471	perform	0.083	symptom swine	1.626
blood	0.05	flu web	-5.463	personnel	-1.359	take care	0.21
boni	0.984	fluid	1.87	physic emotion	6.192	talk friend	-2.518
bulg	-0.733	fuss	0.234	pictur	-0.124	tast	0.08
case swine	4.282	germ	0.111	properti	-0.372	team	-0.044
caution	1.174	ground take	-3.022	radar	0.287	throat	0.251
cellular	-2.072	guilti	-0.394	reach	0.201	time knock	6.523
check code	4.495	habit	0.381	remain	-0.666	tissu	-0.012
check site	0.149	halt	0.819	rough	0.075	total cost	-4.794
checklist	-1.595	health	-0.04	run	0.143	transmit	1.535
chicken	0.286	health care	-0.393	rush	-0.07	troop	0.767
cleav	0.991	healthcar worker	1.339	scari	0.109	underli	-0.221
confirm swine	**21.874**	hippocr	-6.038	seal	-0.091	underli health	11.707
cough	0.234	holidai	-0.021	season	-0.064	unquot	8.753
cough fit	2.395	home wors	6.302	seizur	2.987	upcom	0.071
cough lung	2.406	huge	-0.199	self	0.059	viru epidem	-8.805
cough night	6.748	ion channel	4.974	sik	-0.542	visit doctor	-3.456
cruis	-1.186	irrig	8.721	site	0.06	wave	0.033
daughter	0.048	item	-0.219	sleep well	0.753	weight loss	-0.296
die swine	0.196	kick ass	-0.15	soak	0.41	wikipedia	0.66
dilut	2.708	knock	0.24	sore head	2.023	woke sweat	-19.912
drag	0.147	major	-0.376	speed	-0.198	wonder swine	7.266

As a general remark, keeping in mind that those features have been selected using data containing one significant flu period, they cannot be considered as very generic ones.

Regional inference results are presented on Figures 5.7, 5.8 and 5.9 for classes U, B and H respectively. There is a clear indication that the inferred signal has a strong correlation with the actual one; *e.g.* for feature class H (Figure 5.9) the linear correlation coefficients between the inferred and the actual flu rate for *Central England* & *Wales*, *North England* and *South England* are equal to 0.933, 0.855 and 0.905 respectively. Using all folds of the cross validation, the average linear correlation for classes U, B and H is equal to 0.905, 0.868

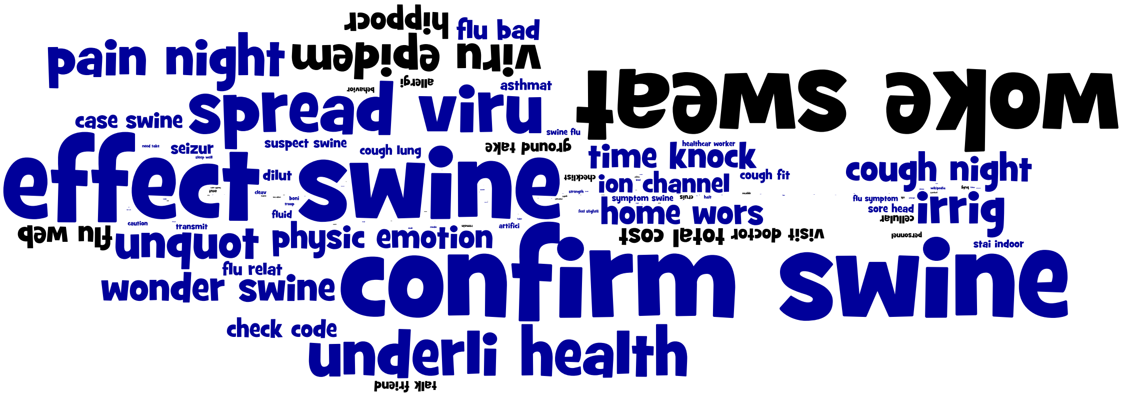


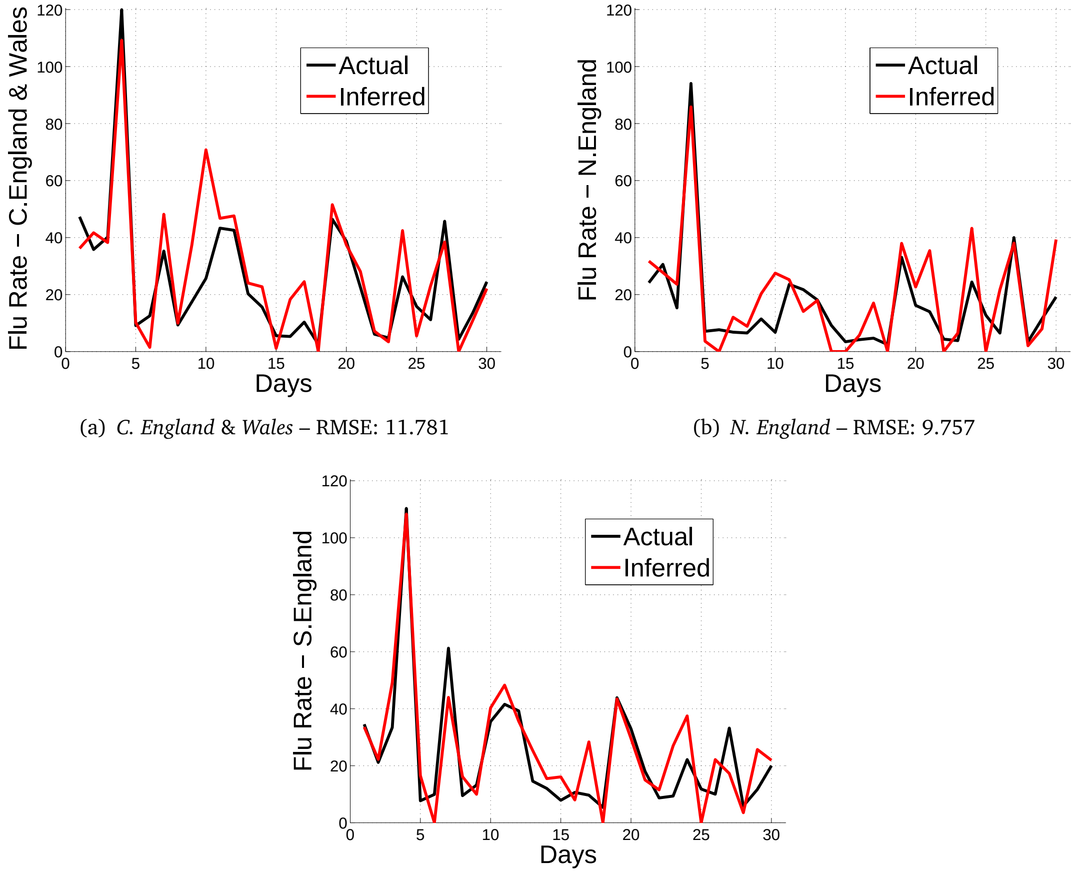
Figure 5.6: Table 5.10 in a word cloud, where font size is proportional to regression's weight and flipped words have negative weights

and 0.911 respectively, providing additional evidence for the significance of the inference performance.[8]

Finally, we present some additional experimental results where training, validating and testing have been carried out in a contiguous time-wise manner. From the 303 days of data, we used days 61-90 for validating CT, 91-121 for testing (from 19th of September to 19th of October, 2009) and the remaining days have been used for training. In this formation setting, we train on data from the Swine Flu epidemic period and then test on a period where influenza existed but its rate was within a normal range. In Figure 5.10, we show the inference outcome for *South England* for three distinctive feature classes (U, B and H).[9] We have also included a smoothed representation of the inferences (using a 7-point moving average) to induce a weekly trend. Class H has the best performance; in this example, class B performs better than U.

[8] All p-values for the correlation coefficients listed are $\ll 0.05$ indicating statistical significance.

[9] Results for the other 2 regions were similar. South England region was chosen because it is the one with the highest population (as it includes the city of London).



(a) *C. England* & *Wales* – RMSE: 11.781

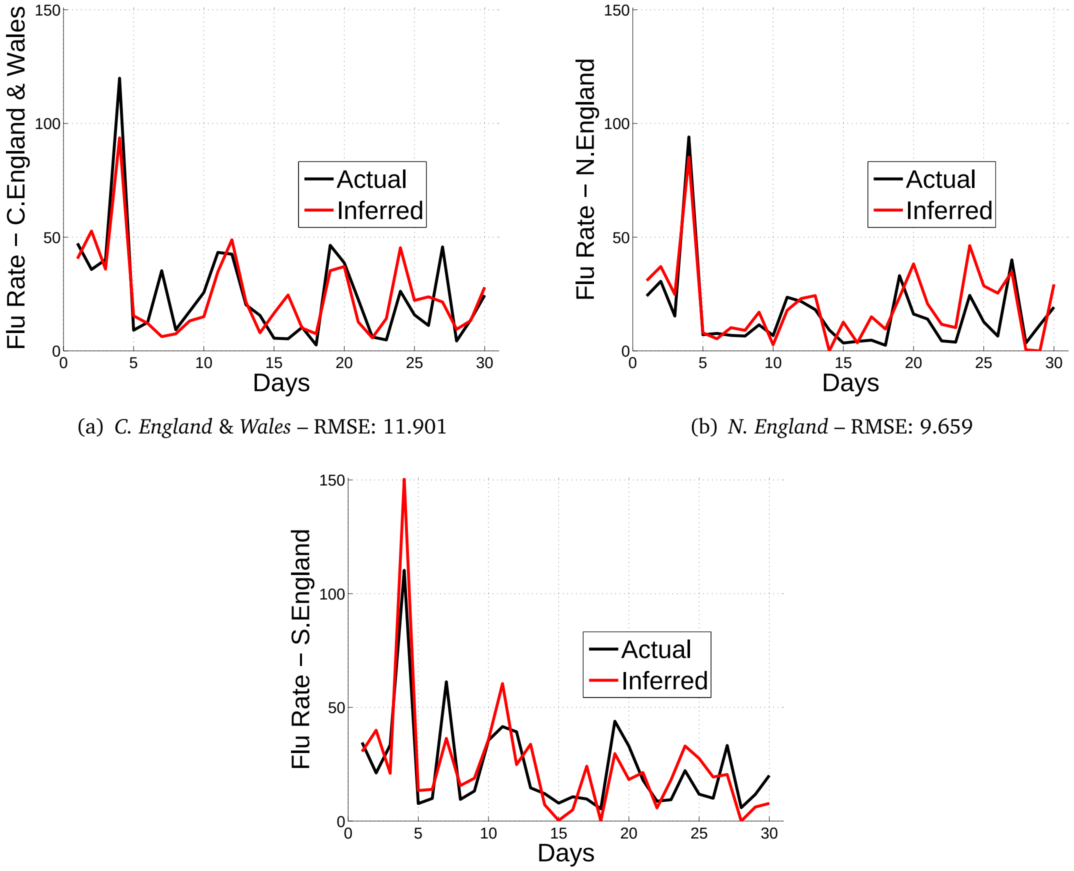
(b) *N. England* – RMSE: 9.757

(c) *S. England* – RMSE: 9.599

Figure 5.7: Feature Class U – Inference for Flu case study (Round 1 of 5-fold cross validation)

5.8 Discussion

The experimental results provided practical proof for the effectiveness of our method in two case studies: rainfall and flu rates inference. Rain and flu are observable pieces of information available to the general public and therefore are expected to be parts of discussions in the social web media. While samples from both rainfall and ILI rates can be described by exponentially distributed random variables (see Section 5.10), those two phenomena have a distinctive property. Precipitation, especially in the UK, is rather unstable, *i.e.* prone to daily changes, whereas a flu rate evolves much more smoothly. Figures 5.11(a) (rainfall



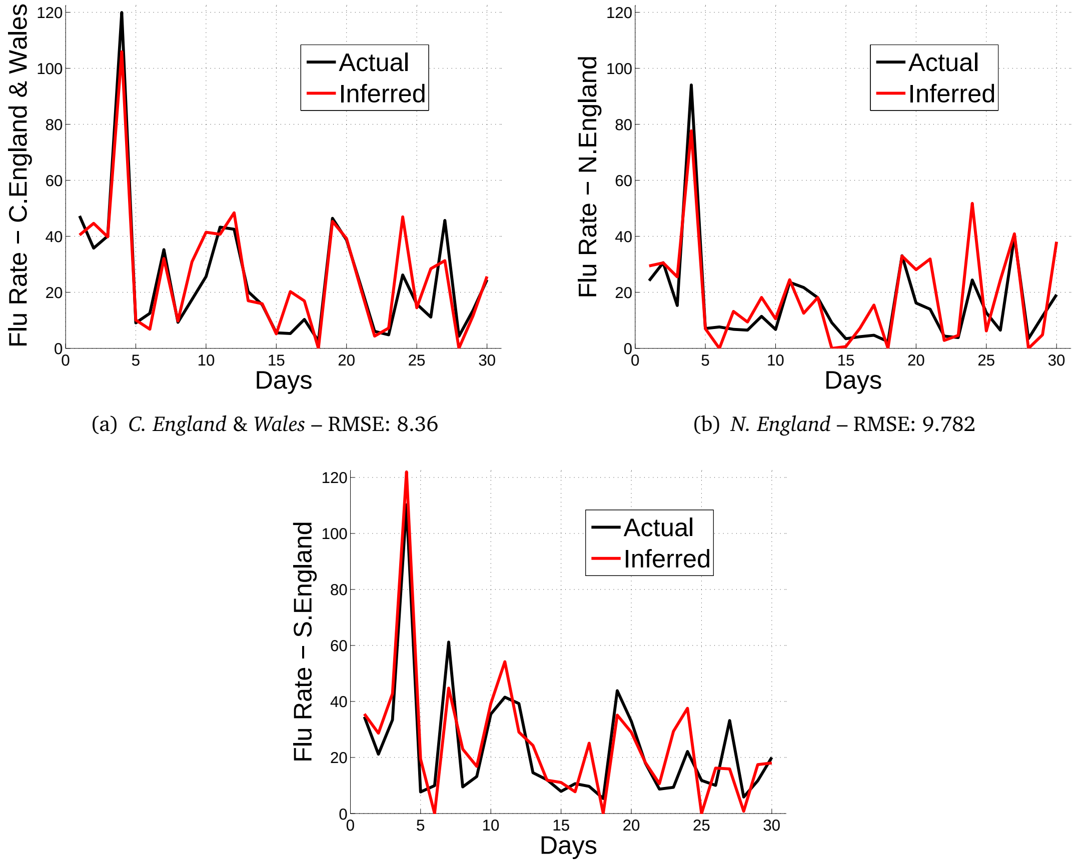
(a) *C. England* & *Wales* – RMSE: 11.901

(b) *N. England* – RMSE: 9.659

(c) *S. England* – RMSE: 13.536

Figure 5.8: Feature Class B – Inference for Flu case study (Round 1 of 5-fold cross validation)

in *London*) and 5.11(b) (flu rate in *South England*) provide a clear picture for this. Consequently, rainfall rates inference is a much harder problem; users discussing the weather of a preceding day or the forecast for the next one, especially when the current weather conditions contradict, affect not only the inference process but learning as well. For example, in the 6^{th} round of the cross validation, where we derived the worst inference performance, we see that in the VSRs of the test set (which includes 67 rainy out of 155 days in total for all 5 locations), 1-gram '*flood*' has the exact same average frequency during rainy and non rainy days – furthermore, the average frequency of stem '*rain*' in days with no rain was equal to 68% of the one in rainy days. Similar statistics are also observed in the training set or for 2-grams – *e.g.* the average frequencies of '*rain hard*' and '*pour rain*' in the training set



(a) *C. England* & *Wales* – RMSE: 8.36

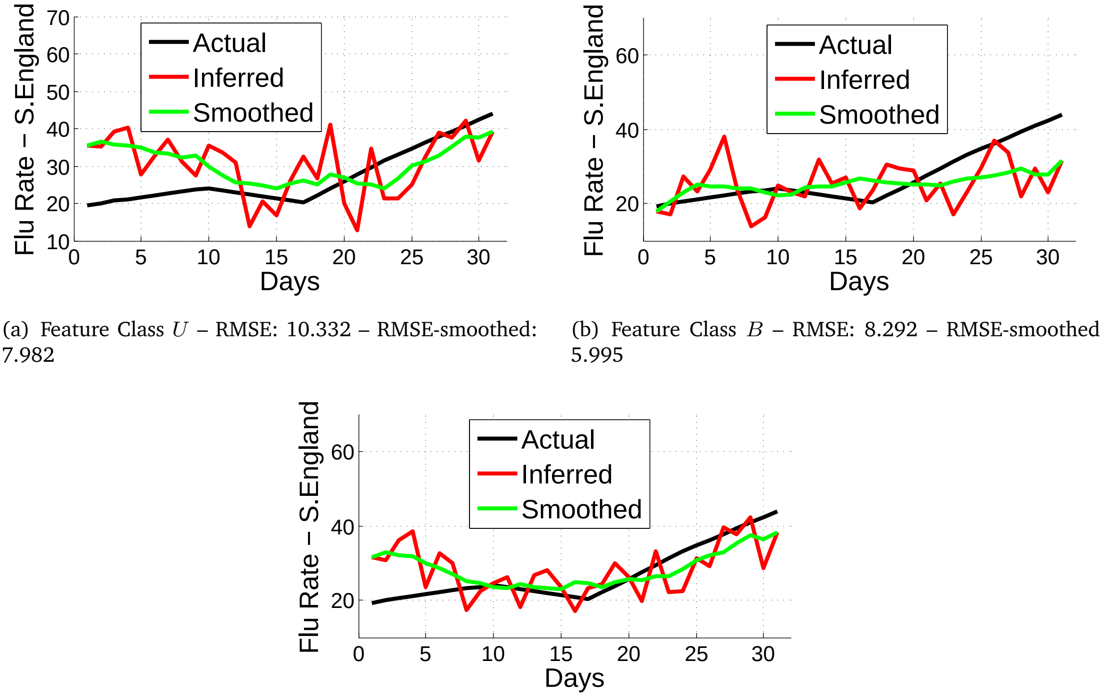
(b) *N. England* – RMSE: 9.782

(c) *S. England* – RMSE: 9.86

Figure 5.9: Feature Class H – Inference for Flu case study (Round 1 of 5-fold cross validation)

(716/1515 rainy days) for non rainy days are equal to 42% and 13% of the ones in rainy days respectively.

The proposed method is able to overcome those tendencies by selecting features with a more stable behaviour to the extent possible. However, the figures in the two previous sections make clear that inferences have a higher correlation with the ground truth in the flu case study – even when deploying a randomly permuted version of the data set, which in turn encapsulates only one major flu period, and therefore is of worse quality compared to the rainfall data. Based on those experimental results and the properties of the target events which reach several extremes, we argue that the proposed method is applicable to other



(a) Feature Class U – RMSE: 10.332 – RMSE-smoothed: 7.982

(b) Feature Class B – RMSE: 8.292 – RMSE-smoothed: 5.995

(c) Feature Class H – RMSE: 7.468 – RMSE-smoothed: 5.537

Figure 5.10: Flu inference results for continuous training, validating and testing sets for *South England* – Testing is performed on data from the 19^{th} of September to the 19^{th} October, 2009

events as well, which are at least drawn from similar distribution functions with flu and rainfall rates (see Section 5.10).

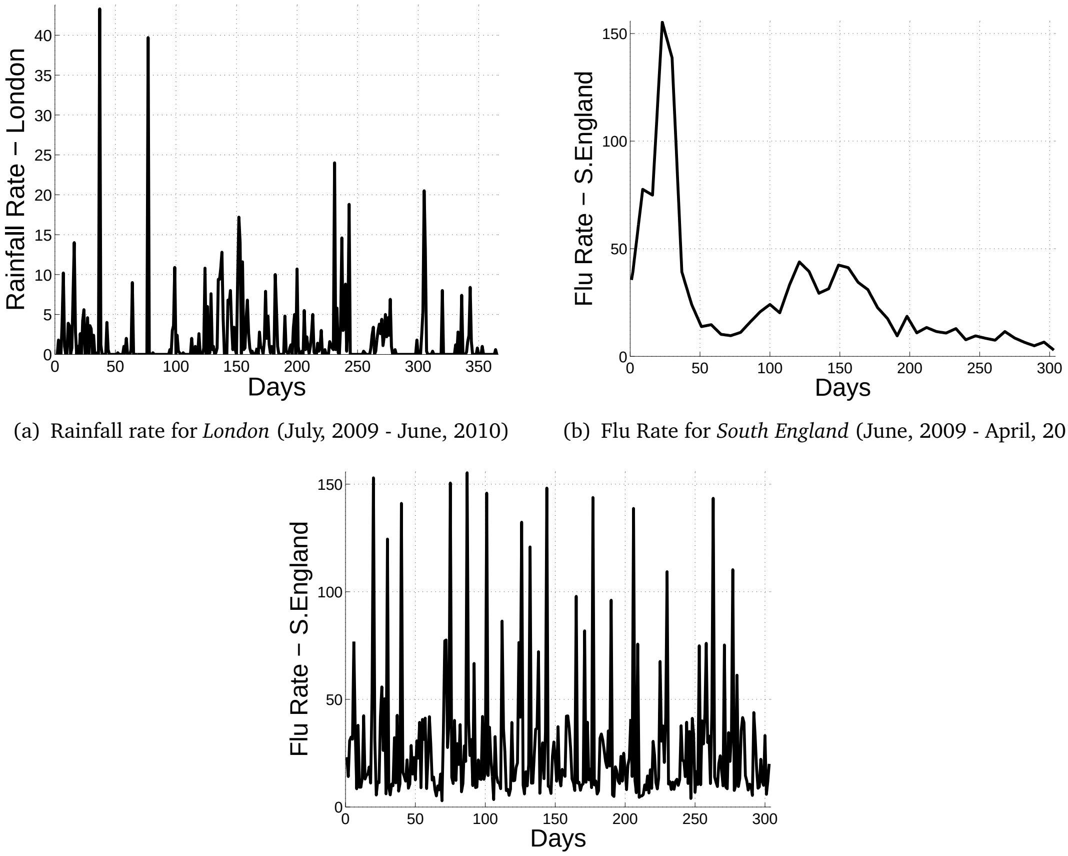
Another important point in this methodology regards the feature extraction approach. A mainstream IR technique implies the formation of a vocabulary index from the entire corpus [104]. Instead, we have chosen to form a more focused and restricted in numbers set of candidate features from online references related with the target event, a choice justified by LASSO's risk bound and the 'Harry Potter effect' (see Sections 4.6 and 4.5 respectively). The short time span of our data limits the amount of training samples, and therefore directs us in the choice of reducing the number of the candidate features to minimise the risk error and avoid overfitting. By having fewer and slightly more focused on the target event's domain candidates, we constrain the dimensionality over training samples ratio and those issues are resolved. Nevertheless, the size of 1-gram vocabularies in both case studies was not small (approx. 2400 words) and 99% of the daily tweets for each location or region contained at least one candidate feature. However, for 2-grams this proportion was reduced to 1.5% and 3% for rainfall and flu rates case studies respectively, meaning that this class of features required a much higher number of tweets in order to properly contribute.

The experimental process made also clear that a manual selection of very obvious keywords which logically describe a topic, such as '*flu*' or '*rain*', might not be optimal especially when using 1-grams; more rare words ('*puddl*' or '*irrig*') exhibited more stable indications about the target events' magnitude. Finally, it is important to note how CT operates as an additional layer in the feature selection process facilitating the adaptation on the special characteristics of each data set. CT's validation showed that a blind application of strict bolasso (CT = 1) would not have performed as good as the relaxed version we applied; only 3 times in the 33 validation sets used in our experiments the optimal value for CT was set equal to 1.

5.9 Applying different learning methods

In the previous sections, we presented a methodology that used soft-Bolasso with CT validation to nowcast events emerging in real life by using Twitter content. This approach is linear since we end up with a set of predictors and their corresponding weights, and in order to make an inference we just have to compute the inner product between their observed values and their weights. In other words the model makes the initial assumption that the relationship between the observations and the response targets is linear in the parameters of the functional form chosen [128]. Moreover, LASSO – as other linear regression techniques – is based on the assumption that either the observations are independent or the responses are conditionally independent given the observations [158].

In this section, we apply an other class of learning functions on the same problem. The motivation behind that is two-fold: firstly, we want to acquire a comparison metric based on different theoretical tools and secondly, we aim to investigate whether our problem is affected by nonlinearities in the data. Thus, for this task we have chosen CART, a decision tree method which is nonlinear and nonparametric, *i.e.* it does not make any assumption on the probability distribution of the data. In total, we have applied three CART methods: the basic CART, its pruned version and finally an ensemble of 'bagged' CARTs. Our attention is



(a) Rainfall rate for *London* (July, 2009 - June, 2010)

(b) Flu Rate for *South England* (June, 2009 - April, 2010)

(c) Flu Rate for *South England* randomly permuted

Figure 5.11: Comparing smoothness of ground truth between the two case studies

drawn mostly on the latter one as it is expected to achieve a better performance; the former are used as intermediate results which assist for a more comprehensive interpretation.

5.9.1 Nowcasting rainfall rates with CART

In this series of experiments we use the same data as in Section 5.6 – *i.e.* tweets, candidate features, locations and rainfall observations from 01/07/2009 to 30/06/2010 – in order to be able to compare the performance of the different learners. The performance of each method is computed by applying a 6-fold cross validation; the process is identical to the one used for measuring the performance of soft-Bolasso with CT validation. Each fold is based

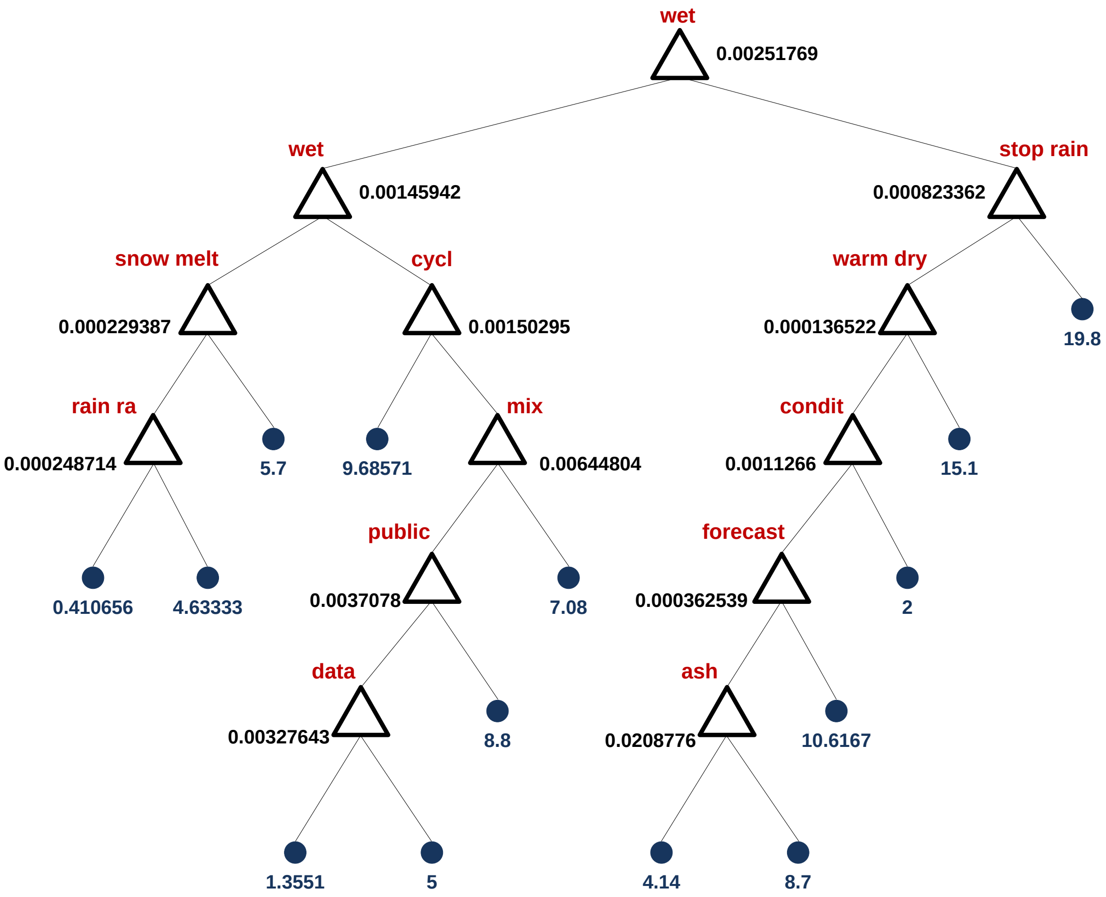


Figure 5.12: CART based on data geolocated in Bristol using the entire data set (365 days) for inferring rainfall rates. This figure only shows a very small fraction of the tree derived by pruning. Numbers (x_i) on each node denote the comparison target for the node's (word's) frequency. If the frequency of the word is larger than x_i, then the next node to visit lies on the right, else we follow the edge on the left side.

on 2 months of data starting from the month pair July-August (2009) and ending with May-June (2010). In every step of the cross validation, 5 folds are used for training, the first half (a month-long data) of the remaining fold for either deciding the optimal pruning level (for pruned CART) or the optimal number of trees (for the ensemble of CARTs), and the second half for testing the performance of the derived learning function.

In total, three feature classes are considered: 1-grams, 2-grams and a concatenation of 1-grams and 2-grams. In our experiments, we have set a lower threshold on the percentage of tree-levels that should be present after pruning. This is equal to 25%, *i.e.* the pruned CART must retain at least the 25% top levels of the full tree. To provide the reader with an

Table 5.11: Rainfall Rates Inference using CARTs (1-grams) – RMSEs for all methods in the rounds of 6-fold cross validation – Fold i denotes the validation/testing fold of round $7 - i$.

Method	Fold 6	Fold 5	Fold 4	Fold 3	Fold 2	Fold 1	Mean RMSE
Bolasso (U)	2.901	1.623	3.04	3.062	2.31	3.443	**2.73**
CART-full	3.784	2.743	5.633	4.65	4.318	4.849	**4.33**
CART-pruned	3.367	2.478	5.549	4.315	4.152	4.705	**4.094**
% of tree-levels (min 25%)	31.4%	25.2%	27.35%	25.79%	36.55%	25.22%	–
Ensemble of CARTs	2.853	1.713	3.395	2.667	2.164	3.487	**2.713**
# of trees (max 150)	43	50	13	93	60	15	–

insight on how an inferred CART looks like, Figure 5.12 shows a fraction of a tree that has been produced by training on all data (1-grams and 2-grams combined) regarding Bristol. We can see that most top words reflect on the topic of weather.

Combining the derivations in [155] and [67] we use 150 bootstraps to form the ensemble of CARTs. In each bootstrap the training set is sampled uniformly with replacement; the number of trees that is going to be used in the ensemble is decided by using the validation set. The ensemble makes an inference by averaging across the predictions of all trees participating in it.

The inference performance of all methods for each feature class in terms of RMSE is presented in Tables 5.11, 5.12 and 5.13. Pruning improves the performance of CART in all cases, but as expected, CART – full or pruned – cannot compete with the more elaborate bootstrap methods. The ensemble of CARTs has a lower RMSE than Bolasso for 1-grams and 2-grams, but when the feature classes are combined, it cannot improve on Bolasso; in that case the error is equal to 2.637, very close to Bolasso's 2.602. In most folds the optimal number of trees used in the ensemble is kept below 50; we also see that the performance of the ensemble of CARTs improves as we add more features.

Feature selection in an ensemble of N trees $\mathcal{T}$ can be performed by first computing an importance factor δ for each variable x_i in every subtree $\mathcal{T}_j$, and then averaging over all the trees in the ensemble [165]. The importance of variable x_i in a tree $\mathcal{T}_j$ is given by

$$\delta(x_i, \mathcal{T}_j) = \sum_{t \in \mathcal{T}_j} \left[\Delta \sum_{i \in t} \frac{(y_i - \bar{y})^2}{N(t)} \right], \tag{5.13}$$

Table 5.12: Rainfall Rates Inference using CARTs (2-grams) – RMSEs for all methods in the rounds of 6-fold cross validation – Fold i denotes the validation/testing fold of round $7-i$.

Method	Fold 6	Fold 5	Fold 4	Fold 3	Fold 2	Fold 1	Mean RMSE
Bolasso (B)	2.745	3.062	2.993	3.028	2.083	3.789	**2.95**
CART-full	4.692	5.066	3.8	3.897	4.14	4.064	**4.276**
CART-pruned	2.98	4.69	3.45	3.148	3.401	3.993	**3.61**
% of tree-levels (min 25%)	26.18%	25.14%	25.24%	29.04%	25.15%	25.62%	–
Ensemble of CARTs	2.865	1.609	3.487	2.816	2.165	3.361	**2.717**
# of trees (max 150)	42	14	6	50	21	30	–

Table 5.13: Rainfall Rates Inference using CARTs (1-grams & 2-grams) – RMSEs for all methods in the rounds of 6-fold cross validation – Fold i denotes the validation/testing fold of round $7-i$.

Method	Fold 6	Fold 5	Fold 4	Fold 3	Fold 2	Fold 1	Mean RMSE
Bolasso (H)	2.779	1.459	2.937	2.954	2.145	3.338	**2.602**
CART-full	3.79	2.698	3.513	3.822	4.69	5.042	**3.926**
CART-pruned	3.494	2.473	3.604	3.717	3.946	4.859	**3.682**
% of tree-levels (min 25%)	25.21%	27.57%	25.56%	29.11%	25.2%	25.22%	–
Ensemble of CARTs	2.758	1.589	3.097	2.732	2.289	3.356	**2.637**
# of trees (max 150)	39	24	149	40	19	31	–

where y_i's denote the observations of node t, $\bar{y}$ denotes their mean, $N(t)$ is the number of observations for node t and Δ expresses the decrease of the enclosed summation as we move down the tree. To compute the overall importance of a variable, we just have to average the above equation over all trees in the ensemble

$$\delta(x_i, \mathcal{T}_{1,\ldots,N}) = \frac{1}{N}\sum_{n=1}^{N}\delta(x^{(u_i)}, \mathcal{T}_i). \tag{5.14}$$

Tables 5.14, 5.15 and 5.16 show the most important features – based on their δ – for round 5 of the 6-fold cross validation in all the investigated feature classes based on the ensemble of CARTs. Those results are directly comparable with the ones in Tables 5.3, 5.4 and 5.5 respectively; the features that have been selected by Bolasso are typed in bold. We see that in all feature classes the distinctively important features are related to the topic of weather; the n-grams '*rain*' and '*rain rain*' are shown to have the largest δ. However, those features are not identical to the ones selected by Bolasso and in some occasions we see that very

Table 5.14: Rainfall Rates Inference using CARTs (1-grams) – Most important features in the ensemble of CARTs (Round 5 of 6-fold cross validation) – δ should be multiplied by 10^{-3} and features in bold have been also selected by Bolasso (U).

1-gram	δ	1-gram	δ	1-gram	δ	1-gram	δ	1-gram	δ
rain	5.308	**umbrella**	0.82	raini	0.336	thunder	0.198	displac	0.166
wet	4.037	**puddl**	0.75	wettest	0.244	lai	0.179	sleep	0.152
rai	1.871	**monsoon**	0.617	**sunni**	0.207	act	0.168		
pour	0.963	sentenc	0.344	manag	0.201	factor	0.167		

Table 5.15: Rainfall Rates Inference using CARTs (2-grams) – Most important features in the ensemble of CARTs (Round 5 of 6-fold cross validation) – δ should be multiplied by 10^{-3} and features in bold have been also selected by Bolasso (B).

2-gram	δ	2-gram	δ	2-gram	δ	2-gram	δ	2-gram	δ
rain rain	3.593	unit state	0.493	long time	0.367	dai week	0.255	**wind rain**	0.195
rain ra	3.358	like rain	0.455	rain lot	0.358	wind direct	0.255	**light rain**	0.192
stop rain	2.484	flash flood	0.429	dai todai	0.318	**horribl weather**	0.249	dai dai	0.178
pour rain	2.346	look like	0.405	start rain	0.316	act like	0.234	oh love	0.172
raini dai	1.062	ic cap	0.393	love weather	0.282	met offic	0.215	rain dai	0.169
love watch	0.513	**sunni dai**	0.384	al al	0.281	climat chang	0.205	rain start	0.165

Table 5.16: Rainfall Rates Inference using CARTs (1-grams & 2-grams) – Most important features in the ensemble of CARTs (Round 5 of 6-fold cross validation) – δ should be multiplied by 10^{-3} and features in bold have been also selected by Bolasso (H).

n-gram	δ	n-gram	δ	n-gram	δ	n-gram	δ	n-gram	δ
rain	4.164	rai	0.72	**pour**	0.39	manag	0.21	white	0.184
rain rain	2.873	**stop rain**	0.627	archiv	0.343	**flood**	0.199	raini	0.174
wet	2.686	sentenc	0.54	**raini dai**	0.26	**puddl**	0.195	wettest	0.172
rain ra	2.542	**umbrella**	0.495	blind	0.258	christma	0.191	japan	0.166
pour rain	1.299	**monsoon**	0.4	solar	0.252	displac	0.187		

irrelevant ones have a relatively high importance such as the 1-grams ‘*sentenc*’ and ‘*archiv*’ in Table 5.16.

5.9.2 Nowcasting flu rates with CART

In this series of experiments we have used the same data as in Section 5.7 – *i.e.* tweets, candidate features, UK regions and official flu rates from 21/06/2009 to 19/04/2010 – in order to be able to compare the performance of the different learners. The performance of each method is computed by applying a 5-fold cross validation; 4 folds using data of 60 or 61 days each are used for training, and from the remaining fold, 30 days of data are used for validation (in order to decide the optimal prune level or number of trees in the ensemble)

Table 5.17: Flu Rates Inference using CARTs (1-grams) – RMSEs for all methods in the rounds of 6-fold cross validation – Fold i denotes the validation/testing fold of round $6 - i$.

Method	Fold 5	Fold 4	Fold 3	Fold 2	Fold 1	Mean RMSE
Bolasso (U)	10.426	8.056	13.29	13.699	10.222	**11.139**
CART-full	13.749	17.924	13.42	15.027	10.368	**14.098**
CART-pruned	13.706	17.832	13.381	13.409	10.281	**13.722**
% of tree levels (min 15%)	48.84%	21.97%	15.79%	17.04%	16.15%	–
Ensemble of CARTs	5.758	6.42	16.659	13.783	5.542	**9.632**
# of trees (max 150)	58	144	44	85	17	–

Table 5.18: Flu Rates Inference using CARTs (2-grams) – RMSEs for all methods in the rounds of 6-fold cross validation – Fold i denotes the validation/testing fold of round $6 - i$.

Method	Fold 5	Fold 4	Fold 3	Fold 2	Fold 1	Mean RMSE
Bolasso (B)	11.806	10.536	16.93	12.958	10.986	**12.643**
CART-full	14.567	14.646	24.852	19.682	13.243	**17.398**
CART-pruned	13.798	12.266	23.451	19.208	13.021	**16.349**
% of tree levels (min 15%)	15.6%	15.17%	18.12%	18.8%	21.74%	–
Ensemble of CARTs	10.152	8.747	20.754	14.241	11.763	**13.131**
# of trees (max 150)	7	26	30	42	126	–

Table 5.19: Flu Rates Inference using CARTs (1-grams & 2-grams) – RMSEs for all methods in the rounds of 6-fold cross validation – Fold i denotes the validation/testing fold of round $6 - i$.

Method	Fold 5	Fold 4	Fold 3	Fold 2	Fold 1	Mean RMSE
Bolasso (H)	9.359	7.971	12.094	13.475	9.93	**10.566**
CART-full	14.409	9.882	17.399	16.001	10.775	**13.693**
CART-pruned	14.445	10.274	17.677	14.822	10.586	**13.561**
% of tree levels (min 15%)	53.03%	16.15%	16.67%	15.79%	17.16%	–
Ensemble of CARTs	6.684	6.132	14.203	13.803	6.191	**9.402**
# of trees (max 150)	19	104	47	24	20	–

and the rest for testing. The same experimental process and settings as in the previous section are applied with the minor difference that we have set the maximum pruning to the 85% (not 75%) of CART's levels.

The inference performance of all methods for each feature class in terms of RMSE is presented in Tables 5.17, 5.18 and 5.19. Again, pruning may have improved the performance of CART in all cases, but this was not enough for competing with the ensemble of CARTs or Bolasso. Bolasso performs better than the ensemble of CARTs in the case of 2-grams, but the best overall performance is derived by the ensemble when combining 1-grams and

Table 5.20: Flu Rates Inference using CARTs (1-grams) – Most important features in the ensemble of CARTs (Round 1 of 5-fold cross validation) – δ should be multiplied by 10^{-2} and features in bold have been also selected by Bolasso (U).

1-gram	δ	1-gram	δ	1-gram	δ	1-gram	δ	1-gram	δ
swine	116.504	swineflu	13.743	lower	4.895	symptom	1.846	hot	1.385
flu	69.77	school	9.84	**team**	3.641	**season**	1.7	**erad**	1.114
site	44.269	**wave**	9.438	tamiflu	3.015	autumn	1.641	warm	1.109
member	24.629	summer	8.609	tour	2.666	fact	1.599	live	1.096
strong	14.701	**run**	5.346	quarantin	2.379	factor	1.386	featur	1.071

Table 5.21: Flu Rates Inference using CARTs (2-grams) – Most important features in the ensemble of CARTs (Round 1 of 5-fold cross validation) – δ should be multiplied by 10^{-2} and features in bold have been also selected by Bolasso (B).

2-gram	δ	2-gram	δ	2-gram	δ	2-gram	δ
swine flu	134.501	confirm case	6.982	**wonder swine**	4.494	**healthcar worker**	1.995
case swine	62.402	need check	6.648	pig flu	3.812	flu kill	1.861
suspect swine	26.794	**confirm swine**	6.608	swine flue	2.942	flu vaccin	1.572
flu symptom	23.463	**underli health**	5.362	**symptom swine**	2.914	light switch	1.333
die swine	12.209	**feel better**	5.09	death toll	2.447	**kick ass**	1.124
check site	12.019	**flu jab**	5.088	unit kingdom	2.242	skin care	1.042
contract swine	9.568	**weight loss**	4.698	**flu bad**	2.072	flu work	1.034

Table 5.22: Flu Rates Inference using CARTs (1-grams & 2-grams) – Most important features in the ensemble of CARTs (Round 1 of 5-fold cross validation) – δ should be multiplied by 10^{-2} and features in bold have been also selected by Bolasso (H).

n-gram	δ	n-gram	δ	n-gram	δ	n-gram	δ
swine flu	121.821	**wave**	10.002	tamiflu	1.85	**boni**	1.346
swine	87.641	**flu symptom**	6.77	**check site**	1.79	**healthcar worker**	1.139
flu	34.451	**run**	4.994	mass	1.765	live	1.133
member	23.675	swineflu	3.961	**season**	1.719	spur	1.096
case swine	15.701	**strong**	3.913	lower	1.618	hayfev	1.091
school	15.463	summer	2.653	hot	1.443		
site	13.971	**suspect swine**	2.52	fact	1.413		

2-grams; the RMSE drops to 9.402 and compared to Bolasso's RMSE (10.556), this is an improvement of 11.02%.

Tables 5.20, 5.21 and 5.22 show the most important features – based on their δ – for round 1 of the 5-fold cross validation in all the investigated feature classes, when the ensemble of CARTs is applied. Those results are directly comparable with the ones in Tables 5.8, 5.9 and 5.10 respectively; similarly to the previous section, the features that have also been selected by Bolasso are typed in bold. Again, the set of markers selected by the ensemble of CARTs has differences compared to one selected by Bolasso; still, the selected features are mostly related to the topic of flu or illness. The most important words are now directly related to the swine flu epidemic (‘*swine*’ and ‘*swine flu*’). In addition, 1-gram ‘*irrig*’ which had the highest weight when Bolasso was applied, is not within the most important features of the ensemble.

5.9.3 Discussion

Overall, we have seen that in most subcases – 4 out of 6 – the ensemble of CARTs outperforms Bolasso. Still, for the rainfall rates inference task, the performance of both learners is quite similar and the best one is achieved from Bolasso’s hybrid class; in the flu rates inference problem, we see that the ensemble of CARTs achieves a significantly better performance than Bolasso.

In terms of feature selection, we may conclude that the two learning functions do not select the same sets of features; for features selected by both learners weighting and/or ranking might also be different with high probability. As far as the semantic correlation of the selected features with the target topic is concerned, we see that Bolasso tends to select more ‘reliable’ features. Particularly for the task of rainfall rates inference, even the hybrid class of the ensemble of CARTs has selected very irrelevant words, such as ‘*sentenc*’, ‘*archiv*’, ‘*blind*’, ‘*manag*’ and so on; that might have been the reason why Bolasso has outperformed it in that case. Nevertheless, the selected 2-grams by both learning methods seem to have a closer semantic correlation with the topics, reconfirming our initial assumptions (see Section 5.1).

Based on the fact that the the ensemble of CARTs gives out an importance ranking of a large amount of features and hence, a feature importance threshold must be defined manually, one might argue that Bolasso offers a more preferable (or natural) way to perform feature selection. On the other hand, as the ensemble seems to outperform Bolasso during the inference process, we could perhaps assume that this problem has nonlinear characteristics which cannot be incorporated in a linear learning function.

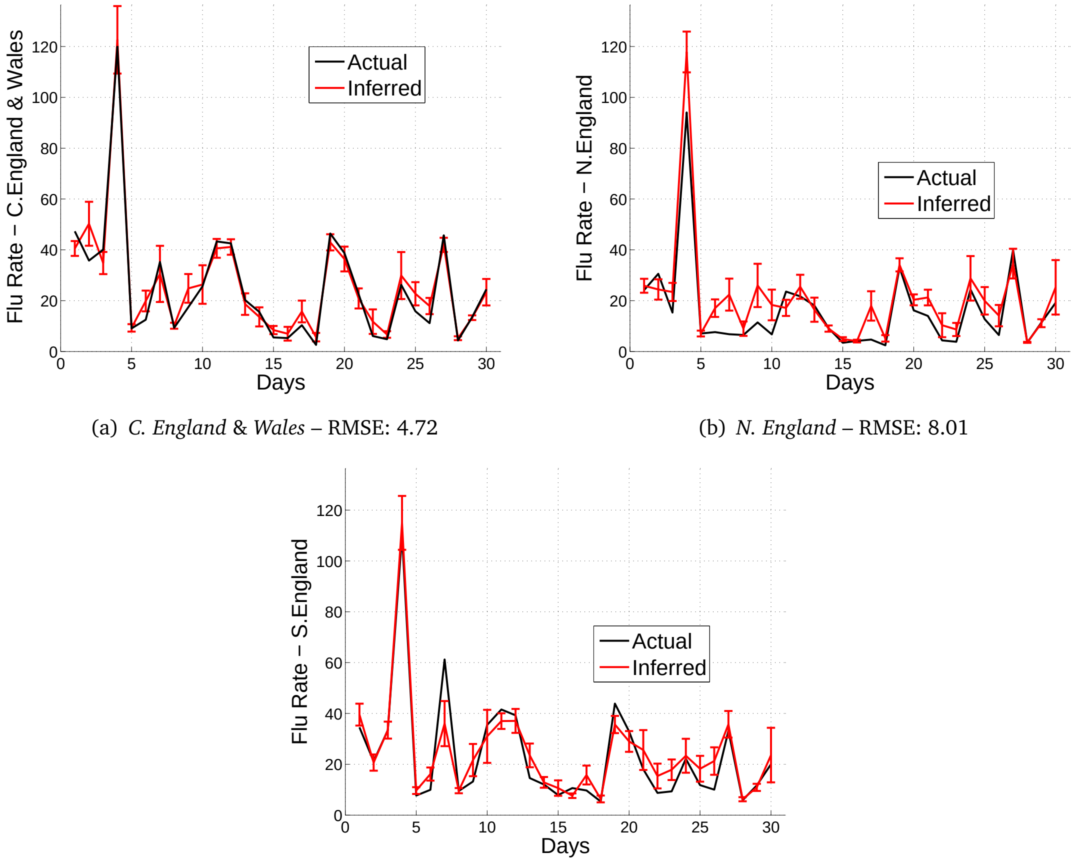
Another statistical fact that might also affect the inference performance of the schemes is the underlying distribution of the ground truth data; in the next section, we are examining this aspect, and show that rainfall and flu rates might be generated from different probability density functions (**PDFs**).

Finally, a general limitation of the inference methods used in our proposed models is that they only produce single inferred values without an associated spread. Spreads on the inferences – usually indicated via confidence intervals (**CI**s) – may assist in identifying locations or regions where our models operate with high confidence and vice versa. One approach for tackling that task could be to approximate the SE of the inferences via the bootstrap samples in the ensemble of CARTs (see Algorithm 2). In this case, SE is equal to the standard deviation of the inferred values of each CART divided by the square root of the number of CARTs in the ensemble. A 95% CI can then be computed by multiplying SE with the .975 quantile of the normal distribution. Figure 5.13 shows inferences with error bars (95% CIs) for Round 1 of the 5-fold cross validation for the flu case study. For the majority of samples CIs tend to be quite tight; we can also observe that CIs across the three considered regions behave similarly for the same daily time intervals (their average pairwise linear correlation is equal to 0.7369).

However, a possibly better way to address this task could be a model based on Gaussian processes. A Gaussian process can be seen as a collection of random variables, where it is hypothesised that any finite number those random variables can be described by a joint Gaussian distribution; it is completely defined by its mean and covariance function [134]. Within the advantages of this framework lies not only the natural way of acquiring spreads for inferences in regression problems, but also the ability of blending various other parameters and information inputs in the model in a quite structured manner (by configuring the covariance function of the Gaussian process). Therefore, temporal as well as spatial characteristics of event detection process could also be explored resulting in probably better and definitely more general solutions.

5.10 Properties of the target events

In this chapter, we have proposed a methodology which is capable of using Twitter content to nowcast rainfall and flu rates in the UK. The purpose of this section is to identify the key properties of those two target events. We assume that the proposed theoretical framework will be – at least – also applicable for inferring the magnitude of other events with similar characteristics. Section 5.8 has already given an initial insight on the main similarities and differences between the random variables of rainfall and flu rates; it has been shown that



(a) *C. England* & *Wales* – RMSE: 4.72

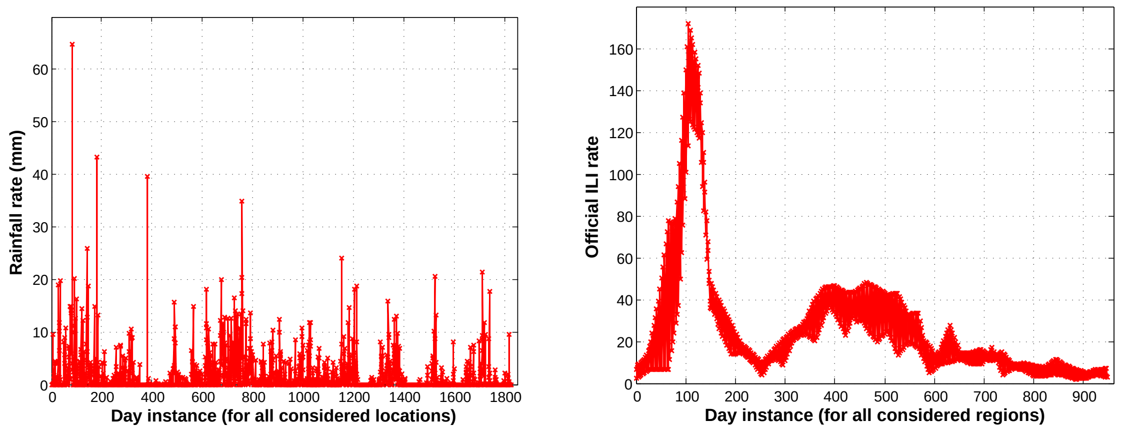
(b) *N. England* – RMSE: 8.01

(c) *S. England* – RMSE: 6.89

Figure 5.13: Inference plots for the flu case study (1-grams & 2-grams) using an ensemble of CARTs (Round 1 of 5-fold cross validation). Error bars depict 95% CIs for the inferred values. This figure is comparable to Figure 5.9.

both are quantities that oscillate, but rainfall tends to do so with a higher frequency, whereas flu rates – in our data set – show a much more smooth behaviour.

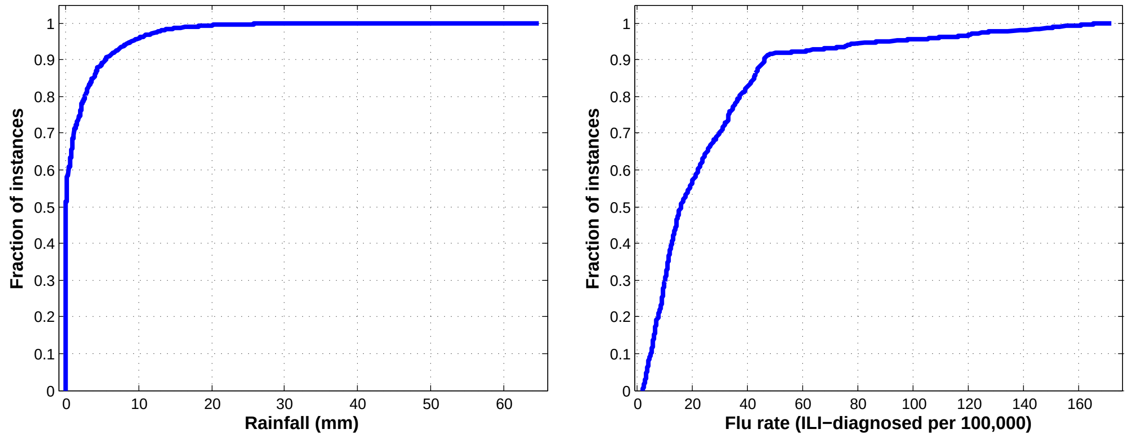
Figure 5.14 holds the plots for the entire sets of ground truth points for rainfall (5.14(a)) and flu (5.14(b)) rates. In each data set we have concatenated the scores of different locations or regions (respectively for rainfall and flu) in a chronological order. Rainfall's mean is equal to 1.7573mm, standard deviation to 3.912mm, the minimum and maximum values are 0mm and 64.8mm respectively and the median is equal to 0mm. Likewise the mean flu rate is 26.0334 (ILI-diagnosed patients per 100,000 citizens), its standard deviation 28.8542, minimum and maximum 2 and 171.9 respectively and the median is equal to



(a) Rainfall rates

(b) Flu rates

Figure 5.14: Ground truth figures by concatenating the data for all locations (rainfall) or all regions (flu) in a chronological order.



(a) CDF for rainfall rates

(b) CDF for flu rates

Figure 5.15: Empirical CDFs for rainfall and flu rates

15.8571. Figure 5.14 reconfirms the argument of the previous paragraph; rainfall rates are – in general – much more unstable than flu rates and many times are equal to 0 (with also a zero median), whereas flu rates are never become 0 in our data set.

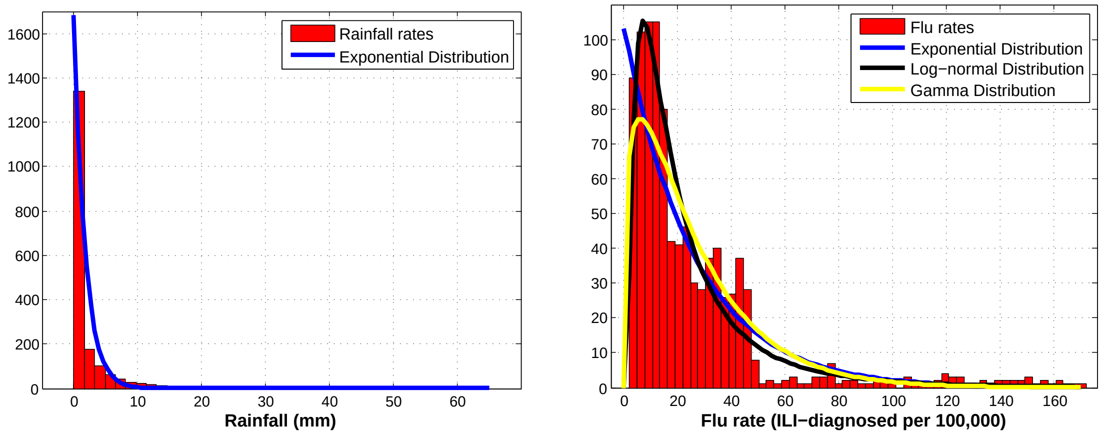
Figure 5.15 shows the empirical cumulative density functions (**CDF**) for the two random variables. For a point 'x' (on the x-axis) the corresponding value of the plot on the y-axis denotes the ratio of the observations which are smaller than 'x' over the total number of observations. From Figure 5.15(a) it is made quite obvious that approx. 60% of the rainfall rates are equal to 0 and 95% is below 10mm. Likewise from Figure 5.15(b) we can see that only 10% of the sample lies between 45 and 172 (ILI-diagnosed patients per 100,000 citizens). Therefore, both events seem to be inactive for the majority of time, and tend to acquire notable values for short periods only.

Finally, we plot the histograms for both variables and try to fit some commonly applied PDFs to them (Figure 5.16). The histogram derived by the rainfall rates points out that the data is most likely to follow an exponential distribution; the best fit occurred for an exponential PDF with $\lambda = 0.569$ (Figure 5.16(a)). However, flu rates – based on the fact that zero or very small values were not the most dominant class – did not show the best fit with an exponential density function. Figure 5.16(b) shows their histogram as well as the fits for three PDFs, an exponential ($\lambda = 0.0384$), a log-normal ($\mu = 2.82451$ and $\sigma = 0.9254$) and a gamma ($\kappa = 1.29098$ and $\theta = 20.1657$). We see that the log-normal PDF offers the best fit with the data, meaning that the logarithm of the samples can be approximated by a normal (Gaussian) distribution.

Summarising the results above, we have seen that our method has been able to track two events that can be approximated by an exponential (rainfall) and a log-normal (flu) distribution. The common ground for both events is that they tend to oscillate – with very different frequencies though –, and that in most occasions and under their metric schemes, they tend to be assigned with small values. Therefore, those targets events are inactive or can be considered as inactive for the majority of time, and become active – or, more precisely, take significant values – for shorter periods of time. As the number of distinguishable peaks increases, we expect the event-tracking process to be more feasible (recall the 'Harry Potter effect' in Section 4.5); by also comparing the semantic correlation with the target topic between the selected features for rainfall (stronger), where more peaks are present, and the ones for flu (less strong), we could argue in favour of such a hypothesis.

5.11 Summary of the chapter

In this chapter, we presented a general methodology for inferring the occurrence and magnitude of an event or phenomenon by exploring the rich amount of unstructured textual information on the social part of the web – in particular, Twitter; our aim was to improve on



(a) Rainfall rates – Histogram vs. Exponential distribution

(b) Flu rates – Histogram vs. Exponential vs. Log-normal vs. Gamma distribution

Figure 5.16: Histograms and PDF approximations for rainfall and flu rates

several aspects of the methodology in [92] and Chapter 4. The specific procedures that we followed, namely soft-Bolasso with consensus threshold validation for feature selection from a large set of candidates or an ensemble of CARTs, have been proven to work significantly better compared to another relevant state-of-the-art approach [56], but the general claim is that statistical learning techniques can be deployed for the selection of features and, at the same time, for the inference of a useful statistical estimator. Several feature classes have been applied, 1-grams, 2-grams and hybrid combinations of them; the combination of 1-grams and 2-grams gave out the best inference performance, whereas the selected 2-grams formed a better semantic representation of the target event.

The overall inference performance of the ensemble of CARTs, which is a nonlinear learning function, was better than the one derived from Bolasso; however feature selection alone was more naturally performed under Bolasso. Comparisons with even more variants of Machine Learning methods are surely of interest, though they would not change the main message: that one can learn the estimator from data, by means of supervised learning. In the case of ILI, other methods (*e.g.* [29, 127]) propose to simply count the frequency of the disease name. This can work well when people can diagnose their own disease (maybe easier in some cases than others) and no other confounding factors exist. However, from our experimental results, one can conclude that this is not an optimal choice.

Furthermore, it is not obvious that a function of Twitter content should correlate with the actual health state of a population or rainfall rates. There are various possible sampling biases that may prevent this signal from emerging. An important result of this study is that we find that it is possible to make up for any such bias by calibrating the estimator on a large data set of Twitter posts and actual HPA readings or precipitation measurements. While it is true that Twitter users do not represent the general population and Twitter content might not represent any particular state of theirs, we find that actual states of the general population (health or weather oriented) can be inferred as a linear or a nonlinear function of the signal in Twitter.

Lastly, some insight on the characteristics of the target events was given. Flu rates are better described by a log-normal PDF, whereas rainfall rates show an exponential behaviour. We hypothesise that our methods are generic and therefore able to nowcast events with similar characteristics.

Chapter 6

Detecting Temporal Mood Patterns by Analysis of Social Web Content

In this chapter, we look into methods that are able to extract affective norms from the content of Twitter. In particular, we are focusing on four mood types, namely anger, fear, joy and sadness and compute seasonal as well as average circadian and daily mood scores. Circadian mood patterns are used to explain behavioural patterns throughout the day; seasonal characteristics are also identified by comparing the patterns of winter and summer. We also show that daily mood patterns can be correlated with significant events or annual celebrations emerging in real-life. This chapter has been partially based on content or ideas from our papers "Detecting Temporal Mood Patterns in the UK by Analysis of Twitter Content" [94][1] and "Effects of the Recession on Public Mood in the UK" [95].[2]

[1] This is a joint work (names of the co-authors are listed in Section 1.3); V. Lampos contributed in all aspects of the research and also wrote the paper.

[2] Based on the fact that V. Lampos was not the first author of [95], we do not cover the exact results and the main topic of this publication, but focus on (as well as extend) the methods and outcomes, where he had a major contribution.

6.1 Detecting circadian mood patterns from Twitter content

Social media, particularly the micro-blogging website 'Twitter', provide a novel way to gather real time data in large quantities directly from the users. This data, which is also time-stamped and geolocated, can be analysed in various ways to examine patterns in a wide range of subjects. Several methods have been already proposed for exploiting this rich information in order to detect events emerging in the population [91], track a possible epidemic disease [93, 148] or even infer the result of an election [164].

Mood seems to change throughout the day, spontaneously or in response to life's vicissitudes. Nonetheless exploring these variations in real time and with large samples is difficult. The use of information gathered via Twitter messaging permits overcoming some of these obstacles. A better knowledge of mood variations is important to confirm widely accepted notions linking certain psychiatric phenomena to diurnal [86] or even seasonal patterns [60]. Recent studies [57, 95] using information captured from Twitter messaging found circadian and seasonal patterns in the content of emotionally loaded wording. As for circadian variation, the volume of words representing negative affect reached its lowest point in the mornings and improved steadily throughout the day. This finding is contrary to the clinical concept of diurnal variation of mood among depressed patients, with the highest load of depressive symptoms early in the mornings. Seasonal patterns, in particular an increase of depressive symptoms in winter months, were not confirmed either [57]. However, this study showed seasonal changes in Positive Affectivity (**PA**) and speculated that "winter blues" may be associated with diminished PA rather than increased Negative Affectivity (**NA**).

We used a similar methodology to gather daily data during two different seasons in our effort to explore circadian and seasonal patterns of mood variation among Twitter users in the UK.

6.1.1 Data and methods

We assessed approximately 120 million tweets, 50 and 70 million for winter and summer periods respectively. For each of the 54 most populated urban centres in the UK, we periodically retrieved the 100 most recent tweets, geolocated to within a 10km range of an urban centre. We attempted to reduce any potential sampling bias by applying the same daily sampling frequency per urban centre. Data was collected for 12 weeks during each season (06/12/2010 to 28/02/2011 for winter and 06/06/2011 to 28/08/2011 for summer). We used a stemmed version of the text in tweets by applying Porter's Algorithm [129]. The tweets were divided into 24 bins, one for each hour of the day, and the emotional valence of each hour was assessed by a text analysis tool. We estimated the level of activity for four emotions – namely fear, sadness, joy and anger – based on the contents of Twitter messages and on emotion-related words in the text.

The emotion-related word-lists were retrieved – a priori – from WordNet Affect [153], a tool that builds on WordNet [111] by selecting and labelling synsets, which represent affective concepts. The word-lists were further processed via stemming and filtering in a way that only single words are kept in the end. After this preprocessing, they were formed by 146 (anger), 91 (fear),[3] 224 (joy) and 115 (sadness) stemmed words respectively (see Appendix B.3). WorNet Affect as well as stemming have been regularly applied in other emotion-detection and text mining applications and thus, are considered as standard ways for this type of information retrieval [22, 154].

6.1.1.1 Mean Frequency Mood Scoring

To extract a circadian (24-hour) pattern for the considered mood types, two approaches – each one based on different assumptions – have been applied. Suppose that n terms are used to track an emotion over $M = D \times 24$ hourly time intervals, where D is the total number of days considered and each time interval is denoted by $t_{d,h}$, where $d \in \{1, ..., D\}$ and $h \in \{1, ..., 24\}$. The collected number of tweets for a time interval is denoted by $N(t_{d,h})$. For every $t_{d,h}$ we count the appearances of each term in the Twitter corpus, $c_i^{(t_{d,h})}$, where $i \in \{1, ..., n\}$. Those 'hourly' term counts are normalised by being divided with the respective number of tweets collected in that time interval and therefore, instead of working with raw counts, we now derive an hourly frequency $f_i^{(t_{d,h})}$ for each term, *i.e.* $f_i^{(t_{d,h})} = \frac{c_i^{(t_{d,h})}}{N(t_{d,h})}$. Next, we compute the mean frequency $F(t_{d,h})$ of the n terms for each time interval $t_{d,h}$:

$$F(t_{d,h}) = \frac{1}{n} \sum_{i=1}^{n} f_i^{(t_{d,h})}. \tag{6.1}$$

[3] The word 'alarm' was removed from the word list for fear, since it showed a strong correlation (summer: 0.8774; winter: 0.8662) with the occurrence of the word 'clock', resulting in a peak when people wake up.

Finally, the mood score $\mathcal{M}(h)$ for an hourly interval h is given by averaging across all days in the data set:

$$\mathcal{M}(h) = \frac{1}{D}\sum_{j=1}^{D} F(t_{d,h}) = \frac{1}{D}\sum_{j=1}^{D}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{c_i^{(t_{j,h})}}{N(t_{j,h})}\right). \tag{6.2}$$

This method is based on the assumption that the frequency of a word indicates its importance. Words with higher frequencies will have a larger impact on the computed means and therefore, in the final mood score. We refer to this approach as Mean Frequency Mood Scoring (**MFMS**).

6.1.1.2 Mean Standardised Frequency Mood Scoring

In the second mood scoring approach, after computing the hourly term frequencies $f_i^{(t_{d,h})}$ ($i \in \{1, ..., n\}$, $d \in \{1, ..., D\}$ and $h \in \{1, ..., 24\}$), we form n vectors W_i of size M which hold the frequencies of each word in all time intervals, *i.e.* $W_i = \{f_i^{(t_{1,1})}, ..., f_i^{(t_{1,24})}, ..., f_i^{(t_{D,24})}\}$, $i \in \{1, ..., n\}$. Next, we standardise W_i's by first subtracting their mean ($\bar{f_i}$) from each entry and then by dividing with their standard deviation (σ_{f_i}):

$$sf_i^{(t_{d,h})} = \frac{f_i^{(t_{d,h})} - \bar{f_i}}{\sigma_{f_i}}, i \in \{1, ..., n\}. \tag{6.3}$$

In that way we 'replace' all frequencies with their standardised versions. The following steps of the method remain the same as before, and hence the 'standardised' mood score $\mathcal{M}_s(h)$ for an hourly interval h is given by:

$$\mathcal{M}_s(h) = \frac{1}{D}\sum_{j=1}^{D}\left(\frac{1}{n}\sum_{i=1}^{n} sf_i^{(t_{j,h})}\right). \tag{6.4}$$

We refer to this approach as Mean Standardised Frequency Mood Scoring (**MSFMS**). MSFMS is now based on the assumption that each mood-word has an equally-weighted contribution to the final mood score.

6.1.2 Experimental results

Figures 6.1 and 6.2 show the 24-hour circadian pattern for the mood types of anger, fear, joy and sadness extracted by applying MFMS and MSFMS respectively. For each emotion type, we have computed the circadian pattern for winter and summer as well as the one of the aggregated data set. In all figures, we have also included 95% CIs using a faded colour. The CIs have been approximated by multiplying the SE of the sample mean with the .975 quantile of the normal distribution, which is assumed to represent the mood scores of each time interval across all considered dates.

Table 6.1 shows the linear correlation coefficients for the inferred circadian patterns between winter and summer under both scoring schemes. Joy is the most correlated emotion in both metrics with correlations higher than 0.92; there is high positive correlation for the remaining emotions as well, pointing out that the overall mood trend does not change significantly between seasons, but might have specific deviations of a smaller impact, which are pointed out and discussed in the following sections.

6.1.2.1 Testing the stability of the extracted patterns

A statistical significance test has been performed on all the inferred average circadian rhythms testing their stability across the daily samples. Consider an emotion type and let its average circadian mood pattern – based on all relevant data – be denoted by a vector v. For every day $i \in \{1, ..., D\}$ in the data set, we compute its circadian mood pattern d_i and then, compute d_i's linear correlation with v, $\rho(v, d_i)$. We form $k = 1{,}000$ random permutations of d_i, compute their linear correlation with v and count the number of times (N_i) that the pseudo-random correlation is higher or equal to $\rho(v, d_i)$. The p-value of this statistical test is, therefore, given by:

$$\mathbf{p} - \mathbf{value} = \frac{1}{D} \sum_{i=1}^{D} \frac{N_i}{k}. \tag{6.5}$$

We consider a mood pattern as stable, when its p-value is below 0.05.

Based on this statistical significance test, the circadian patterns (on the aggregated data sets) derived by applying MFMS for anger, fear and sadness are slightly above the 0.05 threshold, and hence they might not be very stable representations; patterns derived by applying MSFMS (on the aggregated data sets) are all statistically significant. As the proof for instability is not very strong and considering the fact that the scoring schemes are making different assumptions on the weighting of mood terms, we present the extracted signals for both scoring schemes, but investigate further (in terms of peak stability, periodicity and negative versus positive affectivity) only the ones derived by MSFMS.

6.1.2.2 Circadian patterns by applying MFMS

In this section, we describe the circadian rhythms based on MFMS depicted in Figure 6.1. Wee see that fear has a first peak in the morning (7–8a.m.), then drops and from 3p.m. starts increasing again reaching a maximum during 11–12p.m.; the seasonal patterns are highly correlated with the summer rates being slightly higher in all hourly intervals (apart from 7–8a.m.). Sadness has a minimum at 5–6a.m., an intermediate peak at 8–9a.m. and then steadily increases reaching a maximum before midnight; a noticeable difference between the seasons is that winter's morning peak happens one hour earlier than the summer's one and is much stronger. Joy peaks in the morning (8–9a.m.), is decreasing until afternoon (4p.m.), then increases again until 12–1a.m., where it starts to drop reaching a minimum at 4–5a.m.; the circadian pattern is similar among the two seasons, with winter having slightly increased levels of joy. Anger reaches a minimum during the interval 6–7a.m. and then increases until midnight where it peaks; in the summer, we observe increased rates during the late night, while in the winter anger rates are higher in the mornings (7–10a.m.) and during afternoon (2–6p.m.).

6.1.2.3 Circadian patterns by applying MSFMS

MSFMS produces circadian patterns with some noticeable differences compared to the previously presented ones. In Figure 6.2, we see that fear has a first peak at 8–9a.m., then drops until 8p.m. when it starts increasing again reaching a maximum before midnight; we observe higher levels of fear very early in the mornings and during most of the daytime in winter (4a.m.–3p.m.), something that is reversed in the summer between 6p.m. and 4a.m. Sadness peaks in the mornings (8–9a.m.) and then decreases until 5p.m., where it becomes more steady. After midnight, the sadness score is decreasing reaching a minimum at 5–6a.m.; during the winter it is higher between the early morning hours and noon (3a.m.–1p.m.), whereas between 6p.m. and midnight it has higher rates in the summer. Joy also peaks in the morning (9–10a.m.) and then decreases until 9p.m., where it slightly starts to increase again, reaching a significantly smaller peak before midnight; the minimum scores

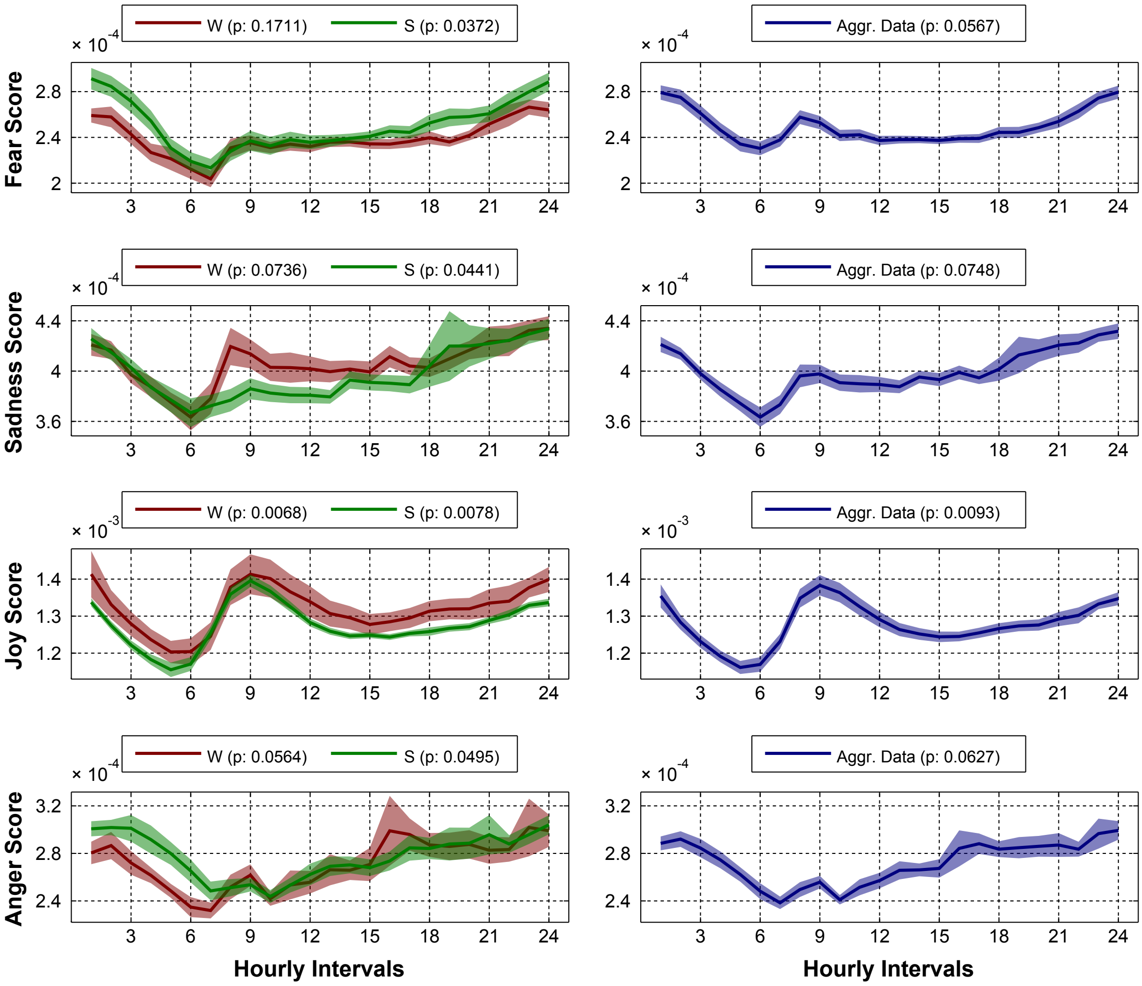


Figure 6.1: These plots represent the variation over a 24-hour period of the emotional valence for fear, sadness, joy and anger obtained by applying MFMS. The red line represents days in the winter, while the green one represents days in the summer. The average circadian pattern was extracted by aggregating the two seasonal data sets. Faded colourings represent 95% CIs on the sample mean. For all displayed patterns the corresponding stability p-values are being reported.

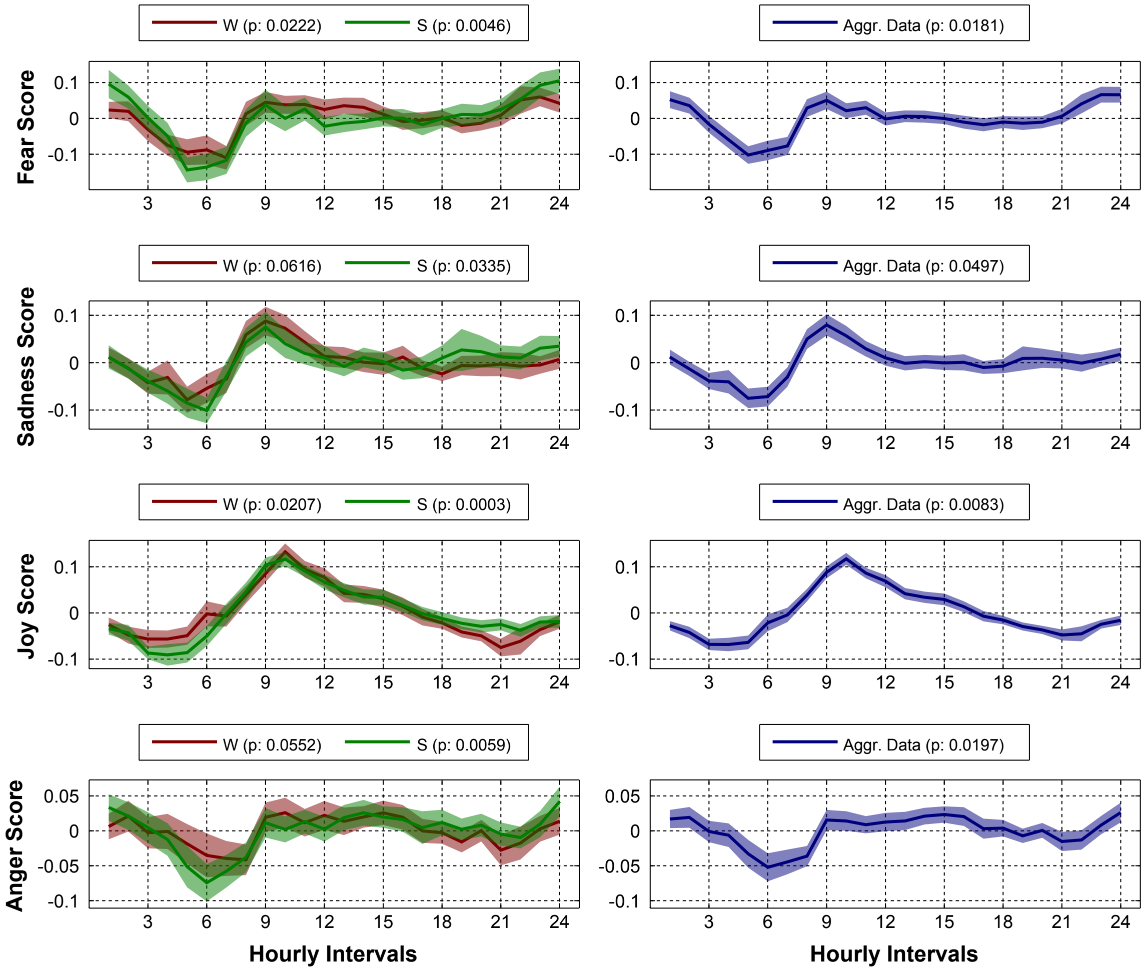


Figure 6.2: These plots represent the variation over a 24-hour period of the emotional valence for fear, sadness, joy and anger obtained by applying MSFMS. The red line represents days in the winter, while the green one represents days in the summer. The average circadian pattern was extracted by aggregating the two seasonal data sets. Faded colourings represent 95% CIs on the sample mean. For all displayed patterns the corresponding stability p-values are being reported.

for joy happen between 3 and 4a.m. The levels of joy during the night hours but before midnight tend to be higher in the summer, however, after midnight 'winter tweets' seem to be more joyful. Anger has a minimum at 5–6a.m., a first peak during 8–9a.m., a second peak at 2–4p.m., then decreases until 9p.m., where it starts increasing again until it reaches a maximum before midnight; in winter during the core daytime we see higher levels of anger, whereas in the summer anger is higher from 4p.m. until the late night hours.

Table 6.1: Winter vs. Summer circadian pattern linear correlations across all the considered mood types for MFMS and MSFMS. All correlations are statistically significant.

	Winter vs. Summer Linear Correlation Coefficients			
	Fear	**Sadness**	**Joy**	**Anger**
MFMS	0.9108	0.7779	0.9598	0.7402
MSFMS	0.8381	0.8309	0.9274	0.7925

Table 6.2: MFMS vs. MSFMS circadian pattern linear correlations across all the considered mood types in winter, summer and on the aggregated data set. Correlations with an asterisk ($*$) are not statistically significant.

	MFMS vs. MSFMS Linear Correlation Coefficients			
	Fear	**Sadness**	**Joy**	**Anger**
Winter	0.722	0.5028	0.3588*	0.2968*
Summer	0.7993	0.3916*	0.6473	0.3813*
Average	0.7091	0.4122	0.5185	0.4211

6.1.2.4 Correlations across scoring schemes and mood types

Table 6.2 shows the linear correlations between the circadian patterns derived under MFMS and MSFMS in winter, summer and on the aggregated data set. All circadian patterns are positively correlated across the two different scorings schemes; some results drawn on seasonal patterns, however, are not statistically significant. Fear and joy show the highest correlations (0.7091 and 0.5185 respectively) and thus, we expect to derive similar patterns independently of the applied scoring approach, something, in fact, displayed in Figures 6.1 and 6.2. Table 6.3 shows the linear correlation coefficients for the circadian patterns of all mood pairs under both scoring schemes. Mood pairs containing the emotion of joy have on average the lowest correlations. Sadness and fear are the most correlated mood types; there also exists some correlation between anger and fear or sadness. The only clearly uncorrelated emotions are joy and anger.

6.1.2.5 Peak moments in the mood signals (MSFMS)

In this section, we investigate further the peak moments of each diurnal pattern extracted by applying the MSFMS scheme. The patterns presented in the previous sections show the average behaviour of each mood type. Some emotions, such as fear and anger, do not

Table 6.3: Linear correlations amongst the considered mood types for MFMS and MSFMS (below and above table's main diagonal respectively). Correlations with an asterisk ($*$) are not statistically significant.

MSFMS/ MFMS	**Fear**	**Sadness**	**Joy**	**Anger**
Fear	–	0.7727	0.3274*	0.6549
Sadness	0.8069	–	0.6976	0.4858
Joy	0.5702	0.5959	–	0.3339*
Anger	0.6154	0.7912	0.0751*	–

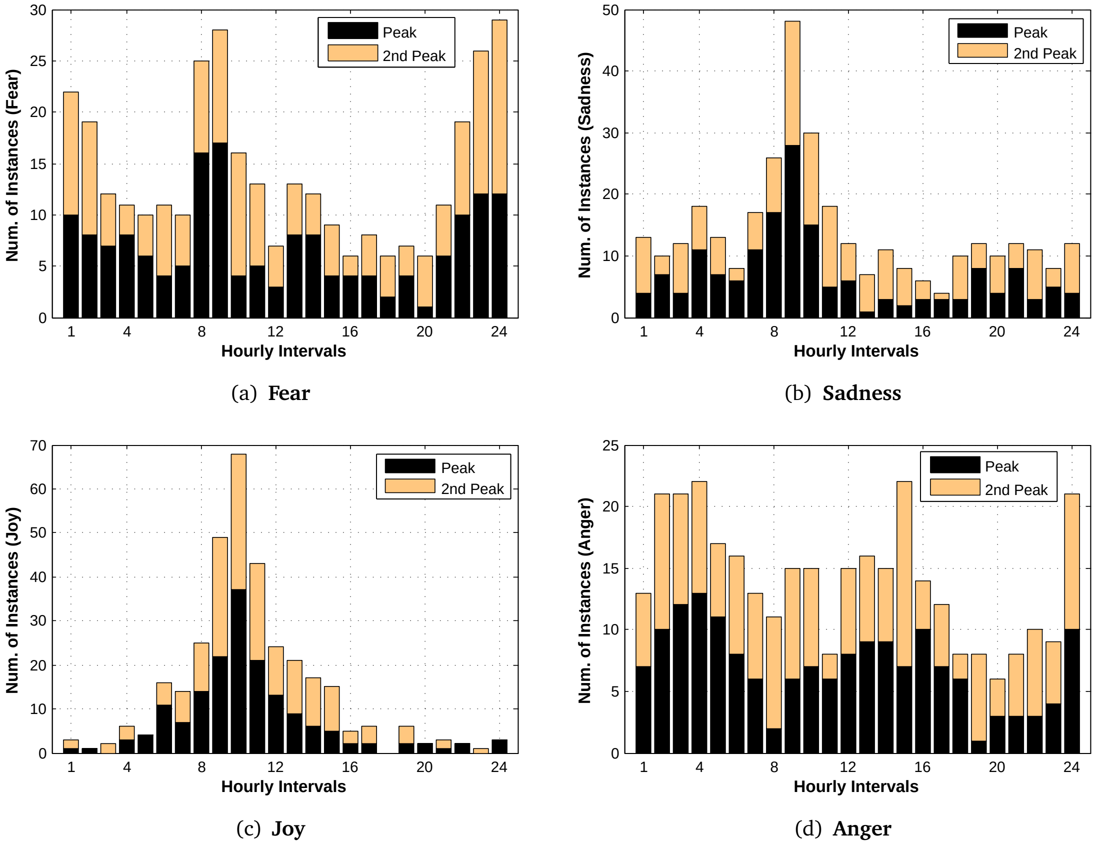
have one clearly distinctive peak in those average patterns and as a result, the stability or importance of the several occurring peaks is unclear. By extracting the peak moments from the daily mood signals in our data set (168 days in total) and forming their histogram, we can observe the distribution of peak moments across all days; this is depicted in Figure 6.3 with the black coloured rectangles for each hourly interval. Since the difference between the highest mood score and the second-highest value might be small in some samples, we have also visualised the frequency of the second-highest mood scores on top of the peak frequency.

We can see that joy (Figure 6.3(c)) retrieves its highest values in the mornings (8–11a.m.) and especially between 9 and 10a.m. Sadness (Figure 6.3(b)) has a similar peaking behaviour; it usually peaks between 7 and 9a.m. The emotion of fear (Figure 6.3(a)) has two peaking time periods; one occurs during late evenings and midnights (10p.m.–1a.m.) and the other one in the mornings (7–9a.m.). Anger (Figure 6.3(d)) has the most unstable behaviour amongst the investigated emotional types. Still, we can derive that it rarely peaks in the evenings, but reaches high levels quite often during midnight, late night hours (1–4a.m.) or right after lunch time (2-3a.m.).

6.1.2.6 Periodicity in mood scores (MSFMS)

As an additional step in our experimental process, we compute autocorrelation figures (see Figure 6.4) for the four mood types based on the MSFMS scheme[4] using lags (ℓ) ranging from 1 to 168 hours, *i.e.* the total number of hours in a week. Consecutive time intervals (hours) are highly correlated for all emotional types; by common logic this is something

[4] The autocorrelations derived by the MFMS scheme were similar, hence, pointing to the same conclusion.



(a) **Fear**

(b) **Sadness**

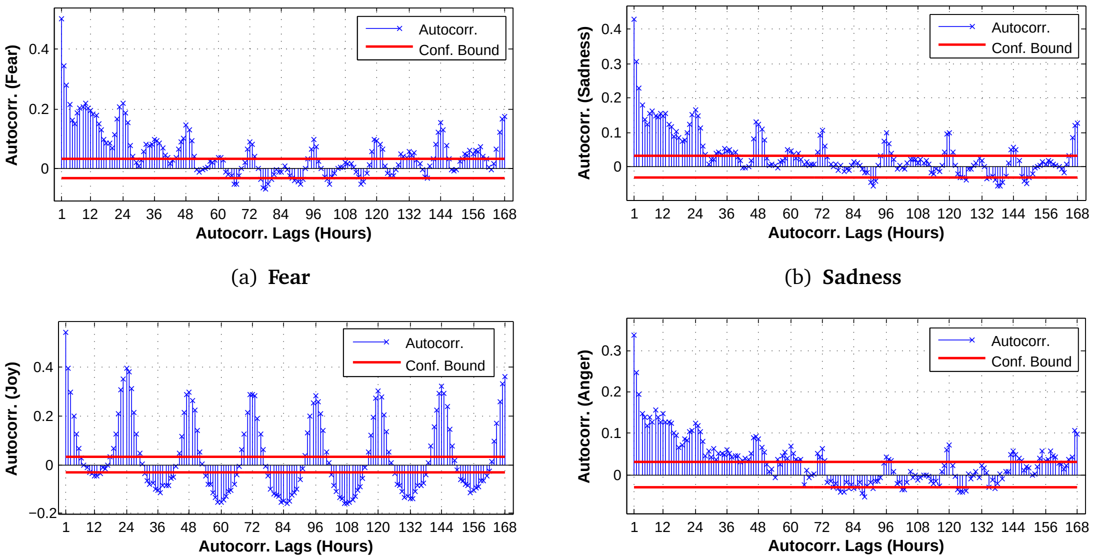
(c) **Joy**

(d) **Anger**

Figure 6.3: Bar plots showing the number of samples which reach a maximum (peak) or second-highest (2nd peak) mood score for each hourly interval and for all mood types when MSFMS is applied.

expected as there is high probability that the mood in the population at a time instance t is affected by the mood in the previous time instance $t-1$. From the autocorrelation plots, it also becomes evident that for each emotional type there exists some level of daily ($\ell = 24$) as well as weekly periodicity ($\ell = 168$). Autocorrelations for $\ell \in \{25, ..., 167\}$ are lower than the one observed for $\ell = 168$ indicating that the weekly pattern (or period) in the mood signal is stronger than the intermediate ones.

We test the statistical significance of the observed levels of periodicity using autocorrelation as the test statistic. Suppose that a vector v of length 4032 (= 168 days $\times$ 24 hourly intervals) holds the scores for a mood type. We compute v's autocorrelations for $\ell = 1$, 24 and 168; then we compute the same autocorrelations for 1,000 randomly permuted versions



(a) **Fear**

(b) **Sadness**

(c) **Joy**

(d) **Anger**

Figure 6.4: Autocorrelation figures for all emotions under MSFMS. The considered lags range from 1 to 168 hours (7-days). Correlations outside the confidence bounds are statistically significant.

of v and count how many times (say k) the levels of autocorrelation for the randomised signal were greater or equal to the ones computed for v. The p-value for this test is therefore equal to $k/1,000$. In our tests for all mood types and considered lags, the computed p-values were equal to 0, indicating that the observed levels of periodicity are statistically significant.

Overall, the emotion of joy has the highest levels of autocorrelation and shows a strong periodic behaviour, whereas periodicity seems to be less strong for the mood type of anger (which also had displayed the most unstable behaviour in terms of peaking times). Those periodicities in affect apart from a generally interesting result, also justify our initial assumption that the extracted mood scores are able to overcome several biases, which might be created, for example, when specific – possibly unfortunate – significant events emerge in real-life and temporarily influence the content of the tweets.

6.1.2.7 Comparison between Positive and Negative Affectivity (MSFMS)

Finally, we merged all mood types expressing NA (anger, fear and sadness) and compared them to the one expressing PA (joy) using MSFMS.[5] Figure 6.5 shows the extracted PA versus NA patterns for winter, summer as well as the entire data set. From all the plots we derive that PA is higher during the daytime, while NA is stronger during the evenings and nights. By comparing winter to summer plots and counting in the seasonal differences between the hours of day and night, we can also see that in winter NA begins to dominate over PA slightly earlier in the afternoons. Winter's most negatively affected hour is 8–9a.m.; in the summer we observe increased levels of negativity during midnight as well.

6.1.3 Discussion

In the previous sections, we performed mood analysis on Twitter content geolocated in the UK trying to extract diurnal – seasonal and average – patterns for the emotions of anger, fear, joy and sadness. As far as we are aware this is the first study of real-time mood variation at a population level using social media information in the UK. We developed two methods for computing mood scores, MFMS and MSFMS; their main difference is based on the fact that the second standardises the frequencies of each mood term over the considered amount of time and thus, reduces any impacts that highly frequent terms might have. Statistical analysis indicates that MSFMS scheme produces more stable results; hence, we show more elaborate findings considering only the latter. Still, the extracted patterns of both scorings schemes are positively correlated with a minimum linear correlation (on the average signals) equal to 0.4122.

Under MSFMS, the mood types of fear, joy and sadness have clear peaks in the mornings. Fear also peaks during midnight; on the flip side, anger has a more ambiguous behaviour with peaks emerging during afternoons, midnights and late night/very early morning hours. The combined NA circadian rhythm shows two peaks, one in the mornings and another one during midnights, whereas PA, which also peaks in the mornings, does so, an hour after NA's peak; the comparison between NA and PA indicates that PA takes higher values in the mornings, but NA increases as the day evolves and takes over PA in the evenings until

[5] In this particular case, MFMS scheme does not, nonetheless, output comparable results due to its dependence on the frequency range of each specific emotion word-list.

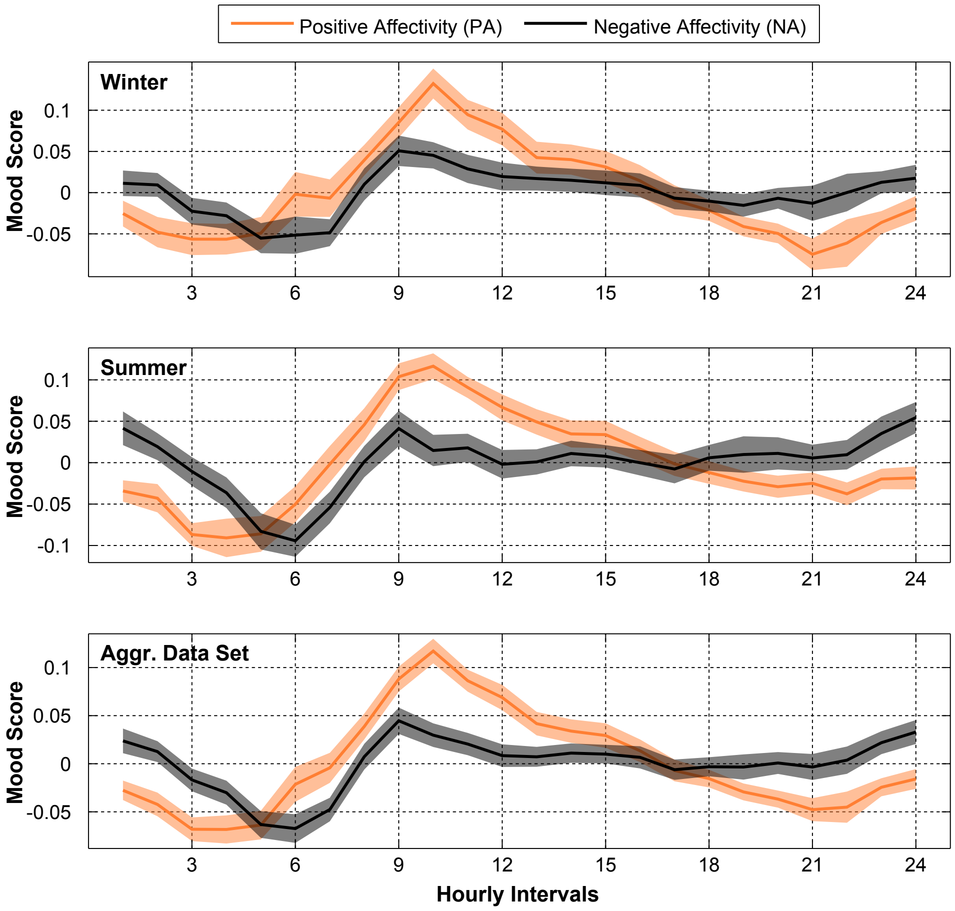


Figure 6.5: These plots represent the variation over a 24-hour period of the emotional valence for Positive versus Negative Affectivity during winter, summer and on the aggregated data set by applying MSFMS scheme. Faded colourings represent 95% CIs. The stability p-values of all extracted patterns are $\ll 0.05$.

the early morning hours. The mood signals are highly correlated across the two different seasons (winter and summer); still there exist some noticeable fluctuations, for example the emotion levels are higher during late night/early morning hours in the winter or there exist stronger emotional indicators during evenings and midnights in the summer.

The comparison of our findings with results from widely reported psychiatric studies reveals some similarities but also several differences. According to [138], NA increases as the day progresses with a peak in the evenings; something reported also in our results for the summer's NA rhythm. Similarly, [117] shows that from afternoon until the late night/early morning hours NA is – on average – stronger than PA. In [66], PA – in terms of 'laughing' or 'socialising' – has a symmetric peak in the mornings, however the circadian patterns for anger – expressed by the action of 'arguing' – and sadness – expressed by 'sighing' – are not very correlated with our results since they show differences in peak moments as well as in monotonicity. Furthermore, in [152] we observe that the emotional state of 'angry' is reaching a maximum at the same time period as the emotion of anger in our findings; the same holds for the emotional states of 'depressed/blue', 'worry' and the emotion of sadness. The contradicting point in this study is that PA oriented emotional states (such as 'happy' or 'enjoy') are reported to have an increasing behaviour through the day.

A recent similar study presented circadian patterns for PA and NA across the different days of the week and their daily averages among different cultures [57]. Their findings for the regions of UK and Australia[6] are in contrast with the ones we derived for the UK alone. PA is shown to retrieve a peak value during midnight; in our work and under both scoring schemes this is happening in the mornings. Moreover, NA peaks during the late night hours, but we have reported reduced levels of NA during this time period of the day (again under both scoring schemes). Importantly, their results vary – significantly – across different cultures; for example, the circadian rhythms for India or Africa indicate a much better fit with the ones we extracted.

There are limitations when conducting these studies. Among these, it is obvious that people cannot send messages whilst asleep and therefore there is a drop in signalling over night. However, we adjusted our results by normalising (MFMS) or both normalising and standardising (MSFMS) to reduce the impact of this effect. In this work, we did not consider the influence that significant events or natural phenomena emerging in real-life might have on the extracted signals and our method for collecting Twitter messages excluded content geolocated in rural UK areas. However, we argue that such biases are usually resolved to a significant level, when working with large-scale samples of data; the periodical characteristics and the statistical significance (especially under MSFMS) provide further proof for the stability of the extracted signals.

On the other hand, limitations arising due to attributes of the general population that might not be present in Twitter users, such as behavioural motives of the elderly, do create unresolved biases; therefore, in the present script, we can only make claims about the population of UK Twitter users and not about the general population. Finally, further validation of the

[6] They do not present diurnal rhythms for the UK alone.

affective words using, for example, labelled data and supervised statistical learning techniques might improve the accuracy of the proposed mood scoring schemes.

6.2 Detecting daily mood patterns from Twitter content

In this section, we are investigating daily mood patterns on Twitter. Similarly to the previous section, we base our affective analysis on WordNet affect [153], and focus on three negative affective norms (anger, fear and sadness) and one positive (joy).

6.2.1 Data and methods

The methods for data collection and processing have already been described in Section 3.2. All collected tweets are geolocated in the UK, and therefore results are partly affected by this. The considered time period for this study is the year 2011; the total number of tweets used in our experimental process is approx. 320 million.

MFMS and MSFMS are applied again, with the only difference that hourly time intervals are replaced with daily time intervals, *i.e.* we are extracting mood patterns on a daily basis. Therefore, using the same notation as in Section 6.1.1, the mood score $\mathcal{M}(d)$ for a day $d \in \{1, ..., 365\}$ is given by:

$$\mathcal{M}(d) = \frac{1}{n} \sum_{i=1}^{n} \frac{c_i^{(t_d)}}{N(t_d)}, \tag{6.6}$$

where $c_i^{(t_d)}$ is the count of term i in the Twitter corpus of day d and $N(t_d)$ is the number of tweets collected in that day. Likewise MSFMS is defined as:

$$\mathcal{M}_s(d) = \frac{1}{n} \sum_{i=1}^{n} sf_i^{(t_d)}, \tag{6.7}$$

where $sf_i^{(t_d)}$ is the standardised frequency of term i for day d.

Recall that MFMS is based on the assumption that the frequency of a word defines its importance (or weight) in the extracted mood signal, whereas MSFMS ‘treats’ all terms equally.

Table 6.4: Correlations between the derived daily mood patterns under MFMS and MSFMS. Correlations with an asterisk ($*$) are not statistically significant.

	MFMS vs. MSFMS	**Smoothed** (7-point MA)
Anger	0.0588^*	0.096^*
Fear	0.8085	0.7888
Joy	-0.0494^*	-0.1699
Sadness	0.5984	0.5663
NA minus PA	0.5814	0.4617

6.2.2 Experimental results

Figures 6.6, 6.7, 6.8 and 6.9 show the extracted daily mood scores under MFMS and MSFMS for anger, fear, joy and sadness respectively. In each plot, apart from the exact scores we have also included their smoothed version (7-point moving average). The exact values allow us to retrieve the dates of peaks in the emotional signal, whereas the smoothed time series reveal a weekly trend of the emotional level and thus, can be used to extract periods with alleviated or aggravated affect. In this section, we try to explain those time series based on the assumption that the affective norms on Twitter might be emerging from events in real-life. It turns out that significant events – as perceived by the common sense – can be tracked and identified within those time series.

Anger. In Figure 6.6(a) (MFMS) we can see two distinctive periods of anger, emerging in October and July respectively. Anger is peaking on the 9th of October, something that could be explained by a series of events that happened on that day, including the death of many civilians in clashes in Egypt, the announcement that the number of jobless people in the UK reached a new record high, Vettel's win[7] and Liam Fox's security scandal. The other peak on the 15th and 16th of July could have been emerged due to the News International phone hacking scandal.[8] Two additional distinctive moments – not periods – in terms of anger, were the 23rd of January (possibly related with the case of Jo Yaetes' murder), and the 29th of April, the day of the Royal Wedding. However, the signal derived by applying MSFMS is quite different; indeed, the linear correlation of the two signals is almost non existent (see Table 6.4). Yet, this uncorrelated signal uncovers even more interesting moments in 2011. The most 'angry' period is now August 2011, with peaks on the 9th and 10th of this month;

[7] This win secured the Formula 1 (F1) title for S. Vettel, a fact that probably displeased the majority of F1 fans in the UK.

[8] Wikipedia's page on this topic, http://goo.gl/47q2Y.

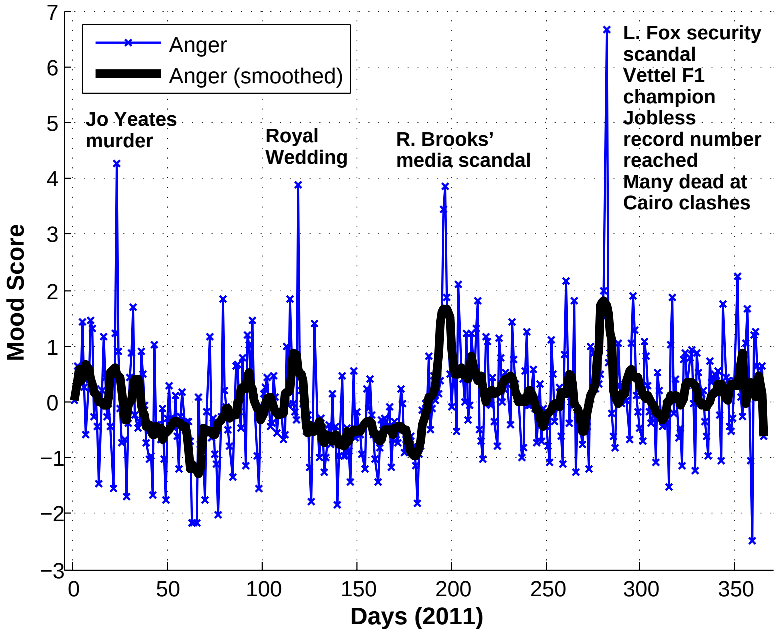
this is when the riots took place all over the UK.[9] The rest distinctive peaks happen on the 18th of January (Jo Yaetes' murder case), 22nd of September (Troy Davis' execution) and the 12th of May (many victims in Syria, Yemen and Libya).

Fear. Fear is the only emotion, where both scoring schemes produce equivalent outputs; the linear correlation between MFMS and MSFMS time series is equal to 0.8085 (see Table 6.4). The two most distinctive fearful moments in both signals (see Figure 6.7) happen during the UK riots (peaking on the 9th of August) and the 9M earthquake in Japan which was followed by a tsunami.[10] Fear is also high on the day of Amy Winehouse's death (23rd of July) which co-occurred with Ander Breivik's attacks in Norway.[11] As a side effect of the wordings used to describe Halloween's traditions, we see that the period of Halloween is also characterised by – possibly – artificial fear.

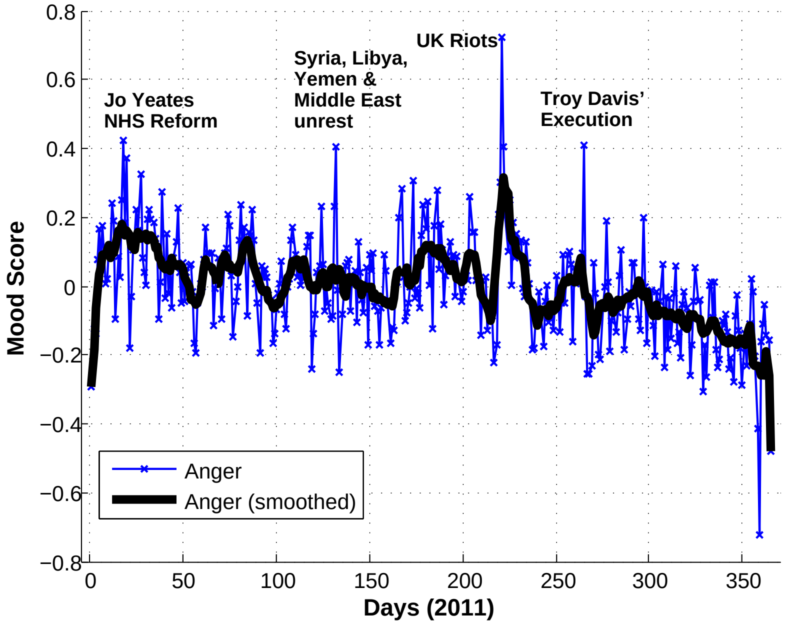
Joy. Similarly to anger, MFMS (Figure 6.8(a)) and MSFMS (Figure 6.8(b)) produce totally different scores for joy and therefore, uncorrelated time series (see Table 6.4). MFMS has the interesting attribute of identifying all significant – at least for the population of Twitter users – events throughout the year; the peak happens on the Christmas Day, and the rest significant days – listed in descending score order – are New Year's Day, Christmas Eve, New Year's Eve, Valentine's Day, Easter Sunday, Father's Day, St. Andrew's Day and Mother's Day. Since MFMS is based on the assumption that the relative importance of each term in the final mood score is defined by its frequency in the corpus, it is made clear that during traditional celebrations the same set of 'joyful words' is overused simplifying their identification. On the other side, MSFMS shows peaks on very different moments, which are much more difficult to explain using calendar events or news outlets. Apart from the peak that clearly correlates with the UK riots (providing evidence that some part of the Twitter population received them positively), the rest could only be explained by major sport events (such as the Cricket World Cup 2011 and the Wimbledon Grand Slam Tennis Tournament), events of international scale such as the Egyptian Revolution or the death of Libyan warlord Muammar Gaddafi, and – possibly – weather conditions.

Sadness. According to Figure 6.9(a) (MFMS), the most sad moments occurred on the 5th of May, a date marked by Osama Bin Laden's death, and the 23rd of July (Winehouse's death co-occurring with Anders Breivik's attacks). The rest significant moments in this time

[9] UK riots time line of events, http://goo.gl/hWF2P.
[10] More information is available at http://goo.gl/C4iHy.
[11] Wikipedia's page on '2011 Norway Attacks', http://goo.gl/C4iHy.

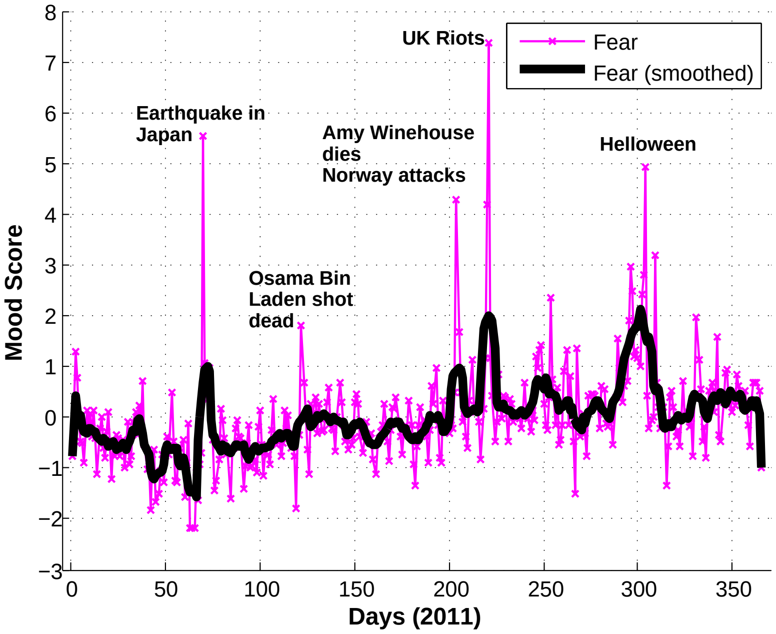


(a) Anger (MFMS)

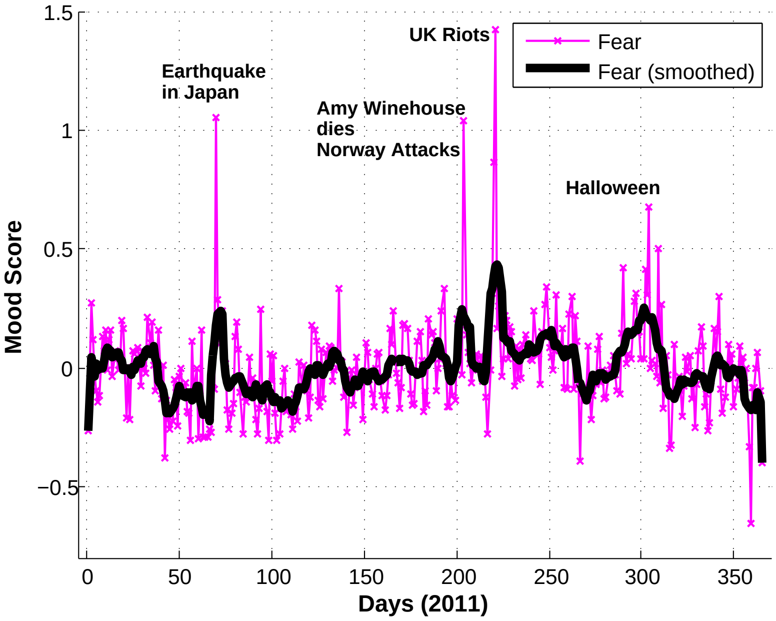


(b) Anger (MSFMS)

Figure 6.6: Daily Scores for Anger (exact and smoothed) in 2011 based on Twitter content geolocated in the UK using MFMS and MSFMS.

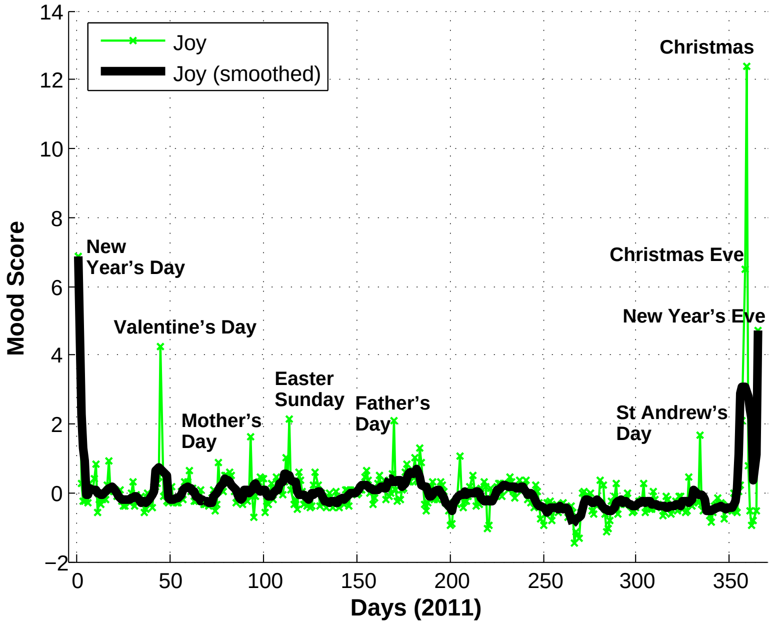


(a) Fear (MFMS)

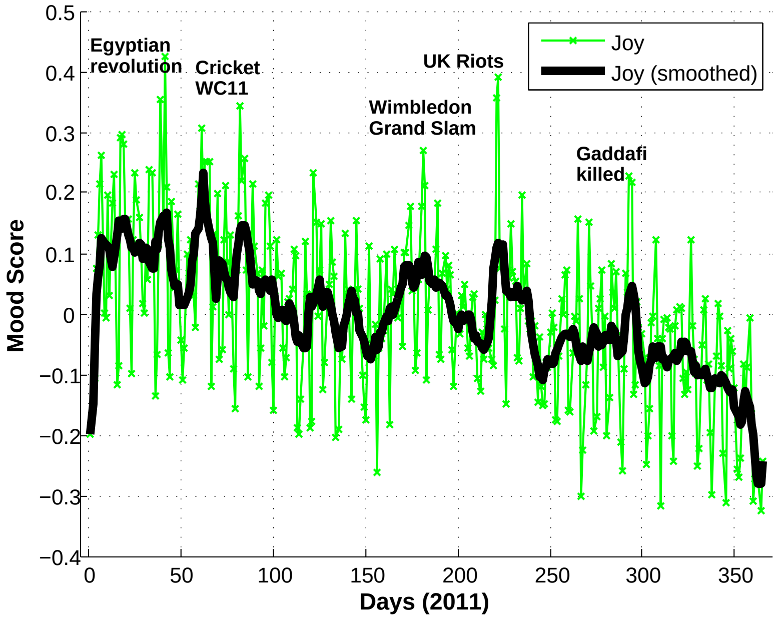


(b) Fear (MSFMS)

Figure 6.7: Daily Scores for Fear (exact and smoothed) in 2011 based on Twitter content geolocated in the UK using MFMS and MSFMS.



(a) Joy (MFMS)



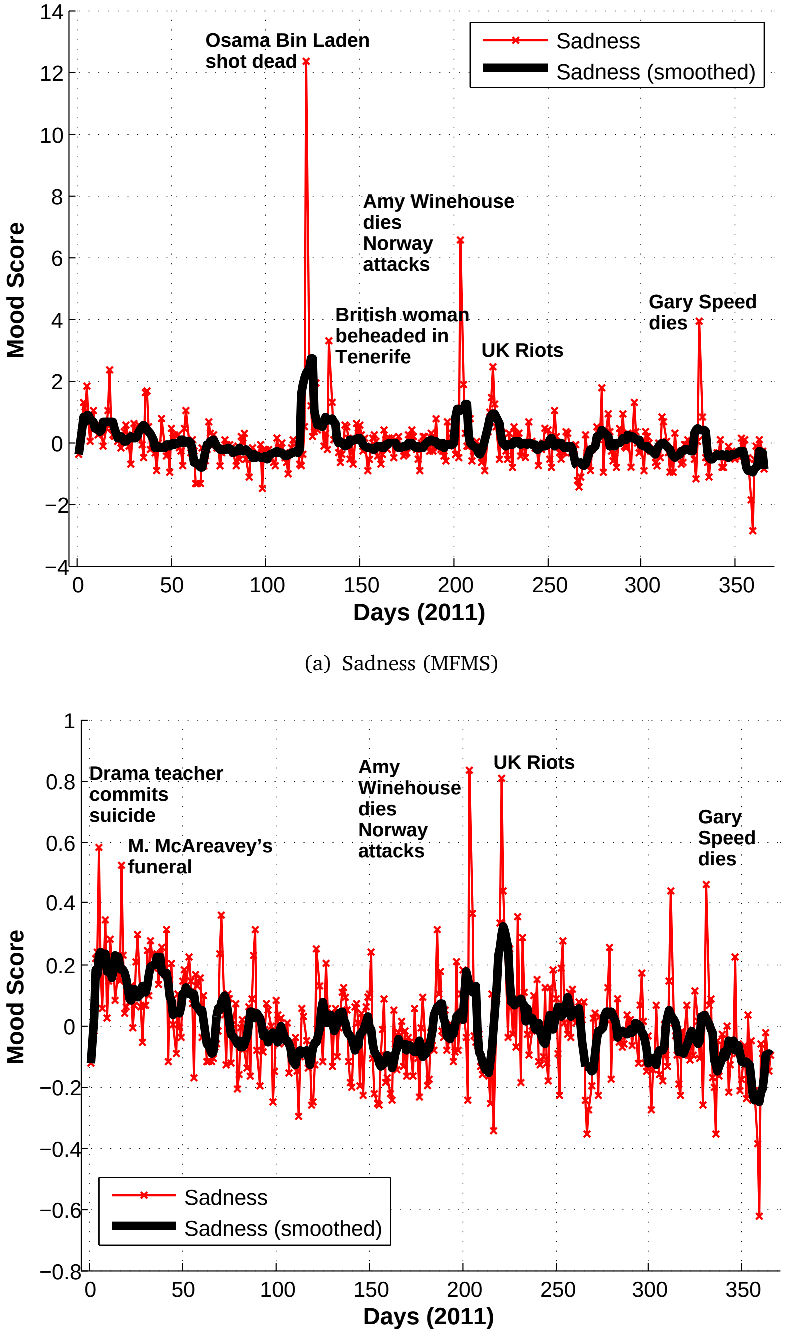
(b) Joy (MSFMS)

Figure 6.8: Daily Scores for Joy (exact and smoothed) in 2011 based on Twitter content geolocated in the UK using MFMS and MSFMS.

series also relate to unexpected deaths (Gary Speed's as well as the beheading of a British woman in Tenerife, Spain). There is also a short peak during the riots in August. MSFMS time series has a correlation of 0.5984 with the result derived from MFMS (see Table 6.4); in those time series, sadness peaks on the 23rd of July, followed by the UK riots period and Gary Speed's death date (27th of November). There are also some other less strong peaking moments that could be matched with sudden or tragic deaths. Overall, in both extracted signals, the main characteristic is that significant moments are usually matched with tragic and/or sudden deaths.

NA minus PA. Figure 6.10 shows the time series of the average NA (the average of anger, fear and sadness scores) minus the score for joy (PA) under MFMS and MSFMS. In that way, we aim to detect days in the year 2011, which triggered all negative emotions of Twitter users. Osama Bin Laden's and Amy Winehouse's deaths together with the UK riots are the most distinguishable events based on the score peaks when applying MFMS; we also see high levels of negative emotion during the earthquake in Japan, the 'Occupy London Stock Exchange' movement, Gary Speed's death and so on. The most non negative day – hence positive – is the Christmas Day. MSFMS time series has a correlation of 0.5814 (see Table 6.4) with the ones of MFMS; the most distinctive dates are now characterised by the UK riots, Amy Winehouse's death and the earthquake in Japan; again the most positive day in the year is Christmas.

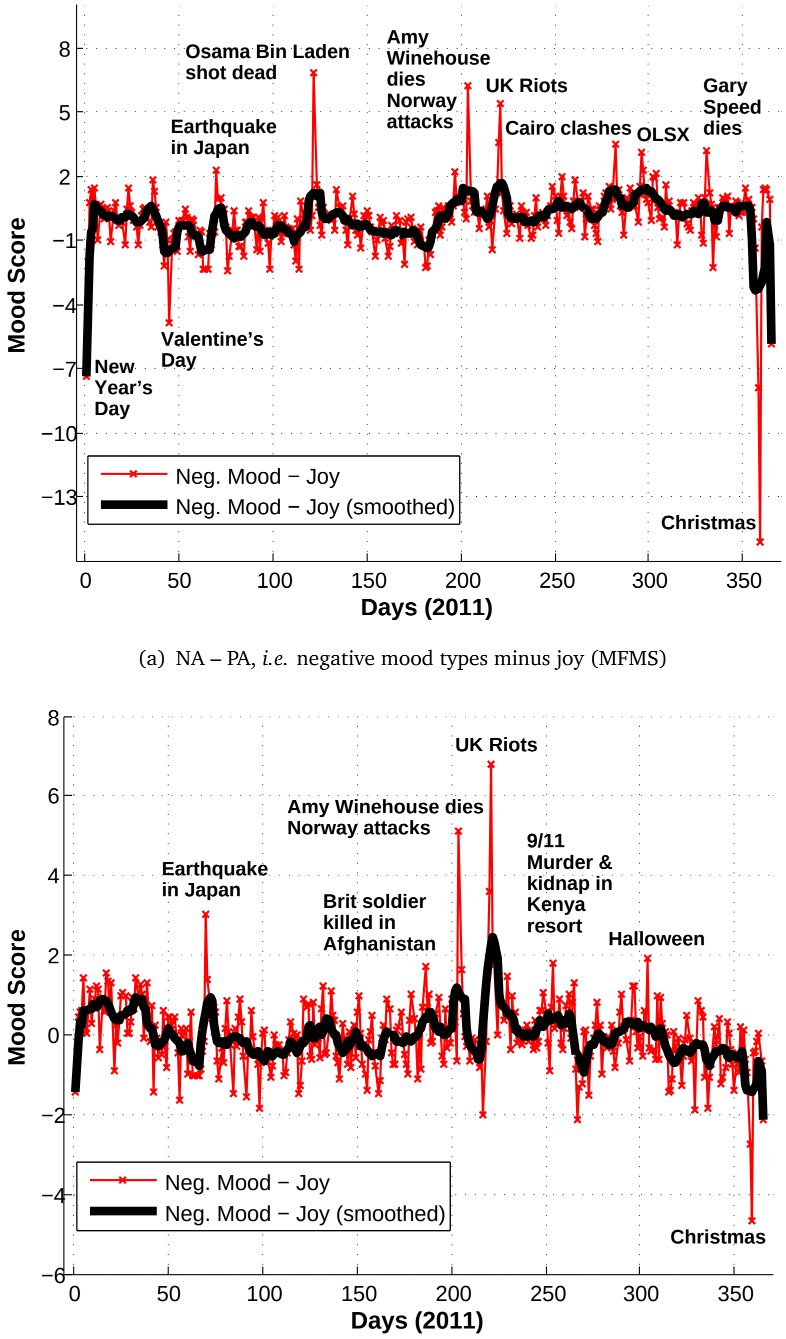
Linear correlations amongst mood types. Table 6.5 shows the linear correlations across the different mood types for the two scoring schemes (below and above the main diagonal for MFMS and MSFMS respectively). We can observe that under MFMS, correlations follow common logic, *i.e.* joy is negatively correlated or uncorrelated with the other emotions; positive correlations exist among anger, fear and sadness, the highest of which occurs between fear and sadness (0.4068). On the contrary, MSFMS shows a very high correlation (0.5528) between joy and anger; still the stronger correlation between negative emotions happens between fear and sadness. Looking at the correlations between the smoothed time series of each emotion (Table 6.6), which reveal similarities in weekly tendencies of each signal, we see that under MFMS the negative correlations of joy with fear and sadness are increasing, whereas the most correlated negative emotions are now anger and fear. MSFMS shows that anger is highly correlated with joy (0.8199) as well as joy with sadness (0.5835); this – at first sight – might seem to contradict with common sense, however those emotional levels do not express the affective norms of an individual, but of a possibly non uniform set of



(a) Sadness (MFMS)

(b) Sadness (MSFMS)

Figure 6.9: Daily Scores for Sadness (exact and smoothed) in 2011 based on Twitter content geolocated in the UK using MFMS and MSFMS.



(a) NA – PA, *i.e.* negative mood types minus joy (MFMS)

(b) NA – PA, *i.e.* negative mood types minus joy (MSFMS)

Figure 6.10: Daily Scores for Negative Affectivity (anger, fear, sadness) minus Positive Affectivity (joy) in 2011 based on Twitter content geolocated in the UK using MFMS and MSFMS.

Table 6.5: Correlations amongst the daily scores of different mood types for MFMS and MSFMS (below and above table's main diagonal respectively). Correlations with an asterisk ($*$) are not statistically significant.

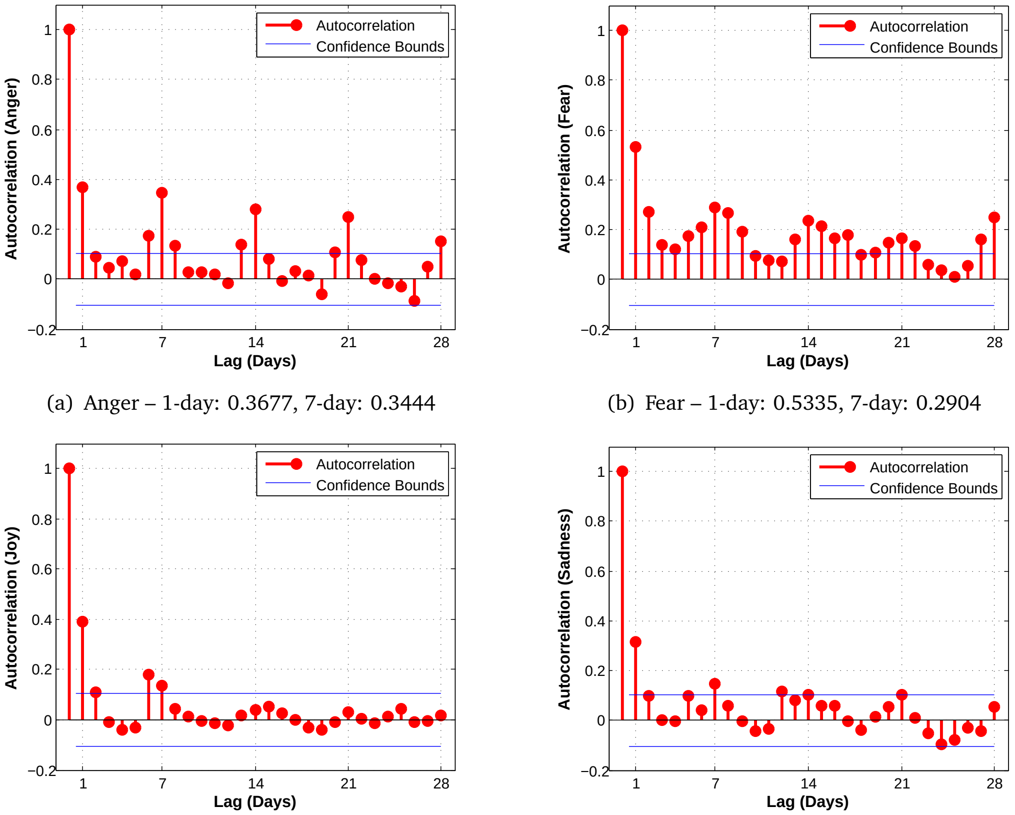
MSFMS/ MFMS	Anger	Fear	Joy	Sadness
Anger	–	0.415	0.5528	0.4686
Fear	0.2654	–	0.1104	0.4772
Joy	-0.0838^*	-0.1322	–	0.2534
Sadness	0.2169	0.4068	-0.1537	–

Table 6.6: Correlations amongst the daily smoothed scores (using a 7-point moving average) of different mood types for MFMS and MSFMS (below and above table's main diagonal respectively). Correlations with an asterisk ($*$) are not statistically significant.

MSFMS/ MFMS	Anger	Fear	Joy	Sadness
Anger	–	0.3684	0.8199	0.6428
Fear	0.321	–	0.0835^*	0.4266
Joy	-0.0796^*	-0.2149	–	0.5835
Sadness	0.1185	0.2203	-0.1812	–

Twitter users. Therefore, our data set can include moments, where opposing emotions may coexist in that population.

Autocorrelation of mood types. Figures 6.11 and 6.12 show autocorrelation figures for up to a 28-day lag (ℓ) (*i.e.* 4 weeks) under MFMS and MSFMS respectively. Correlations inside the region defined by the positive and negative confidence bounds are not statistically significant. Autocorrelation is used as an experimental indication for the existence of periodic patterns in the data (similarly to the experimental process in Section 6.1.2.6). All emotion types and under both scoring schemes show the highest autocorrelations for $\ell = 1$; this may be interpreted by the fact that emotion does not usually change rapidly between consecutive days. MFMS gives the highest 1-day lag autocorrelation to the emotion of fear (0.5335), whereas MSFMS to the emotion of joy (0.5684). The most interesting result here considers the autocorrelations for $\ell = 7$ (the length of a week) as they reveal that mood has a periodic weekly behaviour. Indeed, under MFMS anger, fear and sadness autocorrelations for $\ell = 7$ have the second largest value in the 28 extracted ones; the most strong weekly pattern belongs to the emotion of anger (0.3444). An exception is joy's strongest autocorrelation which happens for $\ell = 6$; this might be the result of having fixed annual celebration dates



(a) Anger – 1-day: 0.3677, 7-day: 0.3444

(b) Fear – 1-day: 0.5335, 7-day: 0.2904

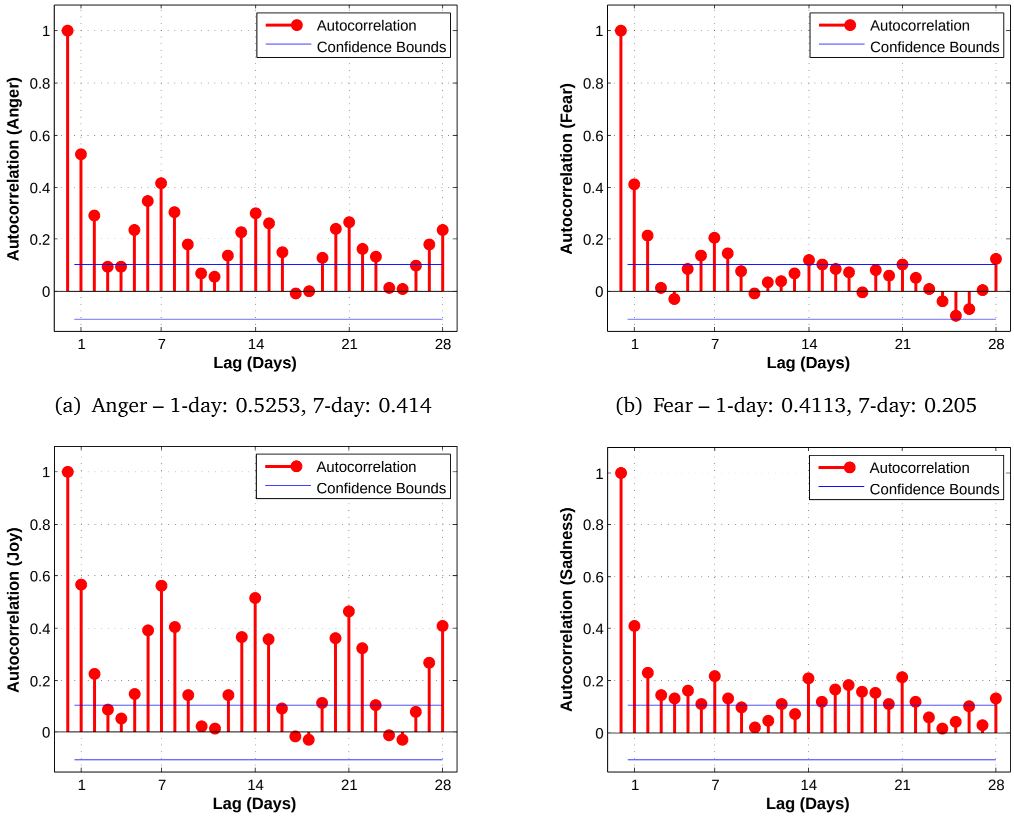
(c) Joy – 1-day: 0.3892, 7-day: 0.1342

(d) Sadness – 1-day: 0.315, 7-day: 0.1485

Figure 6.11: Autocorrelation figures for all mood types under Mean Frequency Mood Scoring (MFMS). Correlations outside the confidence bounds are statistically significant.

which distort the overall periodical pattern. We retrieve very similar results under MSFMS; now all emotions (including joy) have a strong 7-day period with the highest belonging to joy (0.561).

Emotion clusters of weekdays and days in 2011. We average the four mood scores over each weekday in the data set and obtain a 7 (weekdays) × 4 (emotions) matrix, on which PCA is applied. The data set is then projected on the first two principal components and the result for both scoring schemes is visualised in Figure 6.13. Under MFMS we see very clear clusters of consecutive days, *i.e.* Monday-Tuesday, Wednesday-Thursday and Saturday-Sunday. Friday seems to be a ‘special day’ as it has been separated from all other days. The same pattern is evident under MSFMS with the only difference that Tuesday is closer to Wednesday not Monday; this gives an indication that under MSFMS Mondays might



(a) Anger – 1-day: 0.5253, 7-day: 0.414

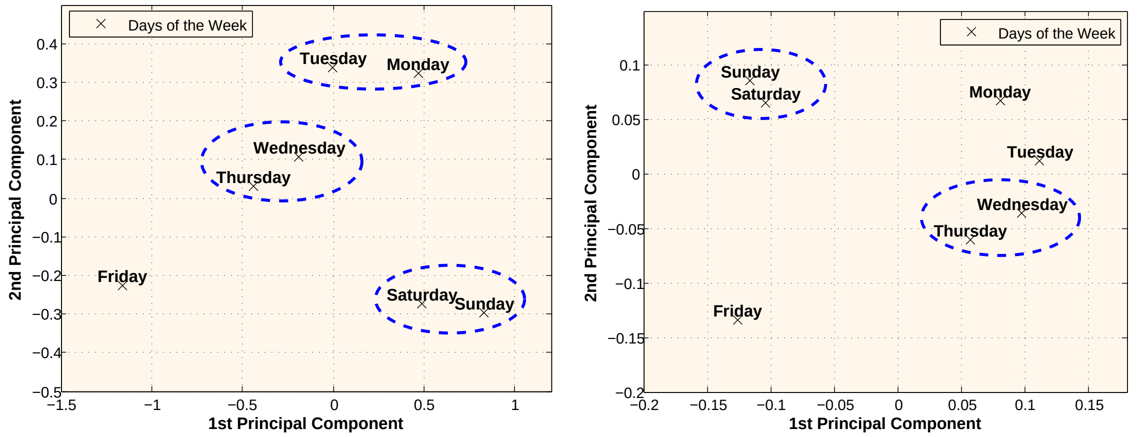
(b) Fear – 1-day: 0.4113, 7-day: 0.205

(c) Joy – 1-day: 0.5684, 7-day: 0.561

(d) Sadness – 1-day: 0.4121, 7-day: 0.2173

Figure 6.12: Autocorrelation figures for all mood types under Mean Standardised Frequency Mood Scoring (MSFMS). Correlations outside the confidence bounds are statistically significant.

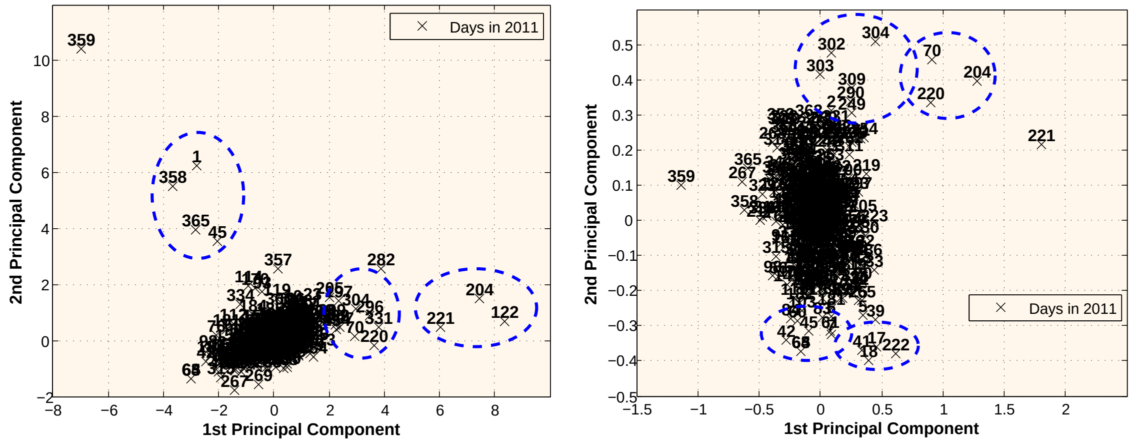
also be represented as days of distinctive emotional patterns. The second task is to cluster single days in 2011 and therefore, PCA is applied on a 365 (days) × 4 (emotions) matrix. The derived clusters (see Figure 6.14) under MFMS and MSFMS – as expected – are not entirely similar. In Figure 6.14(a) we see that the most characteristic moments in 2011 (days 122, 204, and 221 relating to Osama Bin Laden's death, Amy Winehouse's death – Anders Breivik's attacks in Norway and the UK Riots respectively) have been clustered together. Christmas Day (359) is entirely separated from the rest and some popular days of fixed celebrations (New Year's Day and Eve, Christmas Eve, Valentine's Day) have been also grouped together. We also see that the dates of Gary Speed's death, Japan's earthquake and Halloween are also separated from the main cluster of days. Under MSFMS, we observe that Christmas Day (359) and probably the most distinctive day of the UK riots (221) are entirely



(a) Clusters of weekdays using MFMS

(b) Clusters of weekdays using MSFMS

Figure 6.13: Clusters of weekdays based on four mood types (anger, fear, joy and sadness) by applying PCA.



(a) Clusters of days in 2011 using MFMS

(b) Clusters of days in 2011 using MSFMS

Figure 6.14: Clusters of days in 2011 based on four mood types (anger, fear, joy and sadness) by applying PCA.

separated from the rest and have also been placed on opposite points of the considered 2-dimensional space. A distinctive cluster – close to the point ‘221’ – is formed by another date within the UK riots (220) together with the dates of the earthquake in Japan (70) as well as Winehouse’s death combined with Breivik’s attacks (204). The remaining clusters do not deviate much from the main space where the majority of the points lies.

6.2.3 Further discussion of the results

As a general conclusion, we have seen that emotional affect extracted from Twitter's textual stream shows a correlation with real-life events emerging in the UK as well as on an international level. This is something that occurs under both scoring schemes; recall that MFMS is based on the assumption that the frequency of an emotional term defines its importance, whereas MSFMS applies a standardised weighting scheme. Emotional peaks are also present during annual calendar celebrations such as Christmas, Valentine's Day, Halloween and so on. The most significant events of 2011 that concerned the UK, in terms of their correlation with the elevated negative affect on Twitter and under both scoring schemes, are the riots in August, Amy Winehouse's death (combined with the massacre in Norway) and the severe earthquake in Japan.

The two scoring schemes gave rise to quite different events, especially for the emotions of anger and joy; fear and sadness are much more consistent across the proposed schemes. The autocorrelation figures under both scoring schemes indicate that all emotions correlate between consecutive days (lag equal to 1), but when the lag is increased to 2, autocorrelations decrease significantly. This is an indicator pointing out that the state of mood might have a relative 1-day consistency, but usually changes after 2 days have passed. This perhaps is due to 'sudden' events which force emotional reconfigurations. Interestingly, autocorrelations – for all mood types – increase again, when the lag is equal to 7 revealing a weekly periodic pattern. As a result, there is statistical proof that mood is also affected by the day of the week, *i.e.* there is a common emotional ground for each weekday over time. Both scoring schemes indicate that anger might have the strongest weekly autocorrelation (1st under MFMS, 2nd under MSFMS); based solely on MSFMS only, joy has the highest weekly period.

By clustering weekdays based on the two principal components of the 4-dimensional daily affective norm, we discovered that – under both scoring schemes – consecutive days, and especially weekends, tend to group together in a common cluster, apart from 'Fridays' which seem to be unique days by displaying a distinctive emotional behaviour. Finally, by extracting clusters of single dates in 2011, we can observe that all important events are positioned on identifiable areas of the 2-dimensional space and sometimes are clustered together.

Similarly to the previous section, those results face the same limitations. Most importantly, we acknowledge the fact that the users of Twitter might be a biased sample of the population; by using a large data set, we try to overcome or smooth such biases. In contrast to the previous section, where no actual ground truth indications existed (apart from fundamental affective norms extracted by psychiatrists), here we show some evidence – coming primarily by events reported in the news –, which can partly justify moments in the mood signals.

6.3 Summary of the chapter

In this chapter, we investigated seasonal circadian and daily patterns of emotional affect on Twitter content. We considered four mood types, three expressing NA, *i.e.* anger, fear and sadness, and one expressing PA, *i.e.* joy. Two methods have been applied in order to extract mood scores; the first one (MFMS) weighs each emotional term proportionally to its frequency in the corpus, whereas the second one (MSFMS) removes this factor by considering a standardised version of each term's frequency.

Firstly, we extracted seasonal diurnal mood rhythms based on Twitter content published within the geographical region of the UK. The derived results are partly in an agreement with other psychiatric studies, but contradict with the ones presented in a recent similar work [57]. Among other interesting findings, we acquired a clear indication that – within our data set and considering the inevitable biases – PA is stronger than NA during the daytime and vice versa during the night hours. Furthermore, all mood patterns show a level of periodicity; the most common values for the period were 24 and 168 hours (lengths of a day and a week respectively) and the less periodic emotion was the one of anger.

By analysing daily mood patterns, we gained evidence supporting that real-life events affect the mood of Twitter users. By reversing this relationship, mood figures could be used to track significant events emerging in real life; for example during 2011, we saw that the UK riots, Winehouse's death combined with the Norwegian massacre as well the Japanese earthquake or Osama Bin Laden's death affected the emotional output of Twitter users. Among other relevant discoveries, we could also track annual celebrations and calendar events such as Christmas, the New Year's Day, Valentine's, Halloween, the Father's Day and so on. The two different scoring schemes give very different and uncorrelated results for the emotions of anger and joy; however, for fear and sadness – two emotions that are also correlated to each other –, scoring schemes produced quite similar results. Nevertheless, in the analysis of daily mood patterns, we show that the output of the two scoring schemes can be combined as some events identified in one scheme are missed by the other one.

Finally, we showed that all the investigated affective types have a periodic behaviour for lags equal to 1 or 7 days, meaning firstly that consecutive days are more likely to be emotionally correlated, and secondly, that mood might also have a dependency on the specific day of the week. In this direction, we clustered all weekdays based on their mood scores showing that consecutive days are positioned closer on a 2-dimensional space with the exception of 'Fridays'; we also showed that dates with exceptionally negative or positive mood are also identifiable on this space.

Chapter 7

Pattern Discovery Challenges in User Generated Web Content

In this chapter, we present some additional but preliminary work that is mainly based on Pattern Discovery and NLP methodologies and proposes ways to explore further the rich information on the Social Web and in particular, Twitter. At first, we investigate spatiotemporal content relationships between locations in the UK by forming networks based on similarities between their published tweets. We show that those networks are stable over time and therefore, form a trustful description of how content is shared across UK's microblogging space. Then, we briefly examine posting time patterns on Twitter. We show that those patterns are slightly different between weekdays and weekends – something explained by the different characteristics of those time periods – and then use posting time data as features in order to form clusters of days. Finally, a preliminary method is presented where we aim to address the interesting problem of voting intention inference based on inputs from the Social Media; UK 2010 General Election is used as a case study.[1] At this point, we would like to remind the reader that the work presented in this chapter is considered as work-in-progress; still, some results were interesting and could serve as the basis for future research directions.

[1] This work has been published in a technical report titled as "On voting intentions inference from Twitter content: a case study on UK 2010 General Election" [88].

7.1 Spatiotemporal relationships of Twitter content

In the following sections, we investigate spatiotemporal relationships of the content published on Social Networks such as Twitter. One of our aims is to study if and how the geographical distance between locations influences the respective similarity between tweets geolocated in their vicinity. Additionally, we look into whether content originated in different locations but at the same time window has significant similarities, and if such similarities exist, how they are affected by the length of the time window. The latter question is answered with the formation of networks that show how content is shared among locations.

7.1.1 Data

For the following experiments, we are using a set of 71 million tweets published on Twitter during the second semester of 2010 (from July to the end of December) and geolocated in 54 urban centres in the UK. The process of data collection is similar to the one carried out in our previous experiments and has already been described in Section 3.2.2. As a reminder, the radius of each urban centre or location is equal to 10Km, which – in turn – means that all tweets geolocated within this area are mapped to the corresponding location.

We have conducted the same experimental process under two configurations of the settings. In the first experiment, a document is formed by the set of tweets geolocated in one of the urban centres during a day. Since the data set is comprised by 184 days, the total number of documents considered is 184 days $\times$ 54 locations $=$ 9,936. In the second experiment, the time span of a document is reduced to 10 minutes, but we also reduce the total number of days to one month (December). Therefore, the total number of documents becomes 31 days $\times$ 144 documents per day $\times$ 54 locations $=$ 241,056.

7.1.2 Content correlation vs. geographical distance

Is location proximity proportional to content similarities on Twitter? To answer this question we investigate the relationship between those two quantities, *i.e.* the pairwise geographical distance between locations and the respective similarity of their textual content. The pairwise geographical distance is approximated by using the longitude and latitude of the urban centres, and the content correlation is computed using the cosine similarity metric (see Appendix A.3).

We denote the set of the 54 urban centres with $\mathcal{U} = \{u_i\}$, $i \in \{1, ..., 54\}$ and the considered time intervals as t_j, where $j \in \{1, ..., 184\}$ in the first experiment and $j \in \{1, ..., 31 \times 144 = 4464\}$ in the second. A document (set of tweets) geolocated in u_i during t_j is denoted as $\mathcal{D}(i, j)$. The VSR of $\mathcal{D}(i, j)$ is retrieved by applying TF-IDF on the entire collection of documents – after removing stop words and stemming – and is denoted by $d_{i,j}$.

The pairwise cosine similarity of all the documents published during t_j is computed by:

$$\text{Cosine-Similarity}(i, k, j) = \frac{d_{i,j} \cdot d_{k,j}}{\|d_{i,j}\|_{\ell_2} \|d_{k,j}\|_{\ell_2}},\ i \neq k. \tag{7.1}$$

In total we have $\binom{54}{2} = 1431$ location pairs; after removing 15 pairs with geographical distance smaller than 20Km,[2] we end up with 1416 pairs which are denoted by $\mathcal{L}_p$.

For each $\{u_i, u_k\} \in \mathcal{L}_p$ we compute the average cosine similarity over all time intervals t_j:

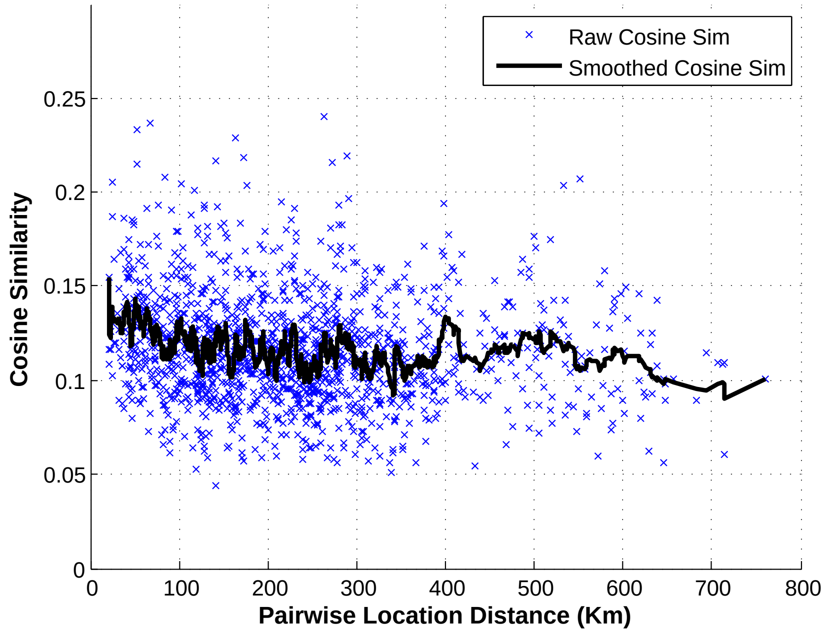
$$\text{Average-Similarity}(i, k) = \frac{1}{n} \sum_{j=1}^{n} \text{Cosine-Similarity}(i, k, j), \tag{7.2}$$

where n is the number of time intervals.

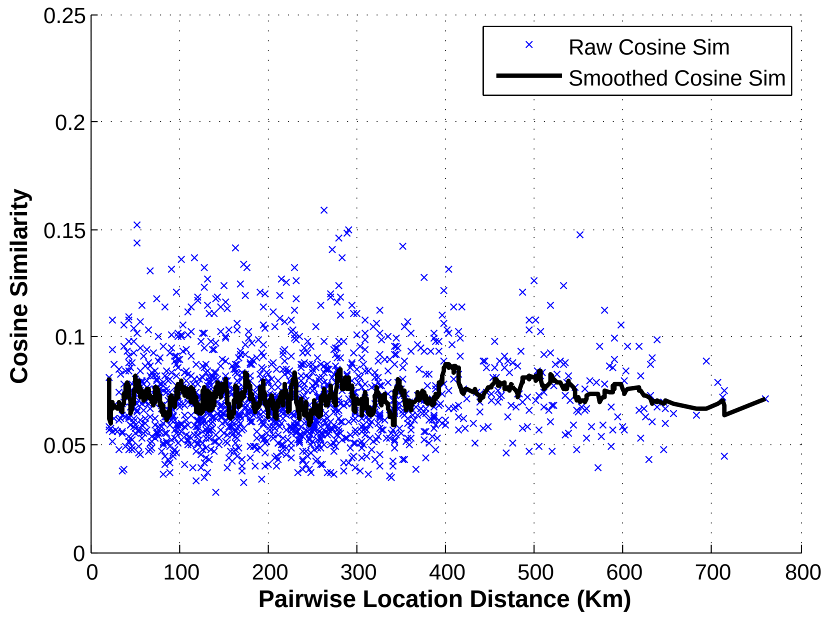
The average cosine similarity for both experimental settings is depicted in Figures 7.1(a) and 7.1(b). By looking at the smoothed equivalents, we see that no obvious relation exists between geographical distance and content similarity; in the first experiment there is a slight decrease in similarity as the distance increases, but this is relatively small to be significant, whereas in the second experiment similarity seems to be unaffected by the distance. In both figures, we can observe high content correlations for urban centres located 500 Km from each other. In addition, the linear correlation between the two results is equal to 0.9185 with a p-value $\ll 0.001$, showing that the time windows used do not affect much the final pattern. However, it also becomes apparent that a decreased time window produces lower cosine similarities.

Therefore, an answer to one of our primary questions is that location proximity does not necessarily affect content similarity on social networks such as Twitter. Of course, looking at

[2] Those location pairs might share identical tweets given that the radius of each urban centre is 10Km.

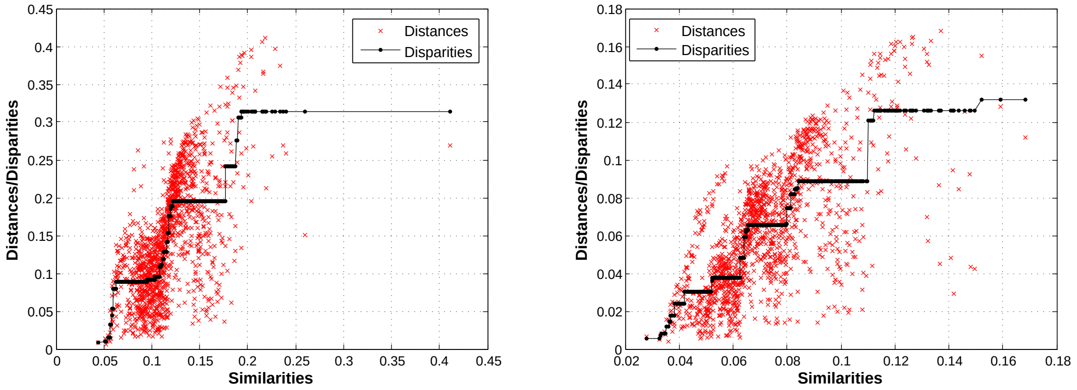


(a) Document's time span: 1 day



(b) Document's time span: 10 minutes

Figure 7.1: The average pairwise cosine similarity for all location pairs plotted against their respective geographical distance. The x's denote the raw points, whereas the black line is their smoothed equivalent (20-point moving average).



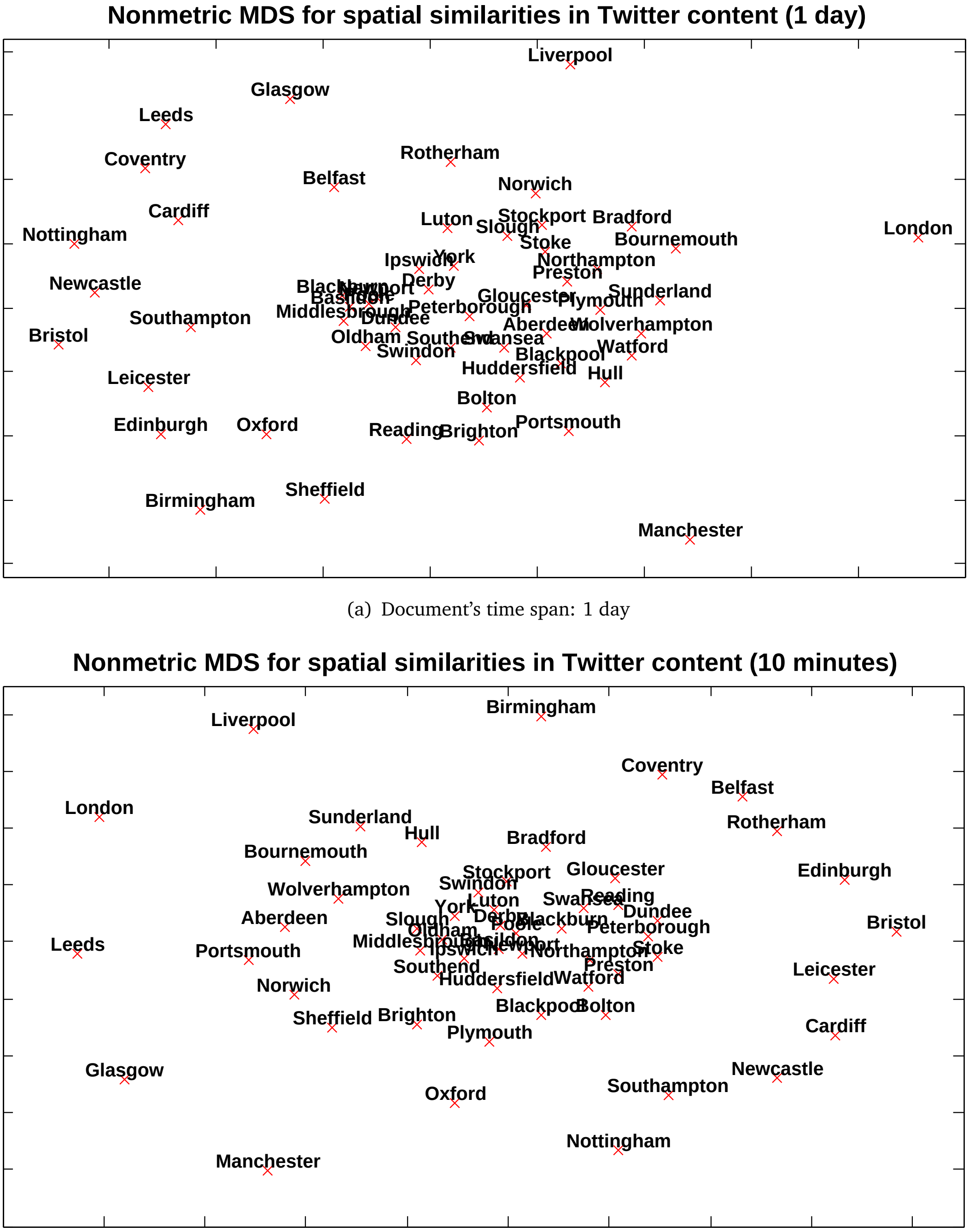
(a) Document's time span: 1 day

(b) Document's time span: 10 minutes

Figure 7.2: Shepard plots showing the pairwise similarities and their monotonic transformation (disparities) when applying nonmetric MDS.

Twitter data on a global level, this pattern is very likely to change. We cannot generalise this result due to the fact that we are using Twitter content geolocated in the UK only. On the other hand, this could also be one of the main characteristics of Twitter and microblogging in general: the ability to quickly spread and share information between places regardless of their actual geographical proximity, at least within the well defined space of a country or a unitary state (such as the UK).

To strengthen our argument and visualise this result better, we apply Multidimensional Scaling (**MDS**) on our data sets, aiming to plot the considered locations on a 2-dimensional space based on their pairwise cosine similarities. MDS is a well-established method for visualising patterns in proximities among a set of variables; after performing several trials on our data set, the best solution was retrieved by performing nonmetric scaling, which is a form of MDS, where the computed solution represents ordinal properties of the data [13]. Figures 7.2(a) and 7.2(b) are the Shepard plots showing the pairwise similarities as well as their monotonic transformations induced by nonmetric MDS, known as disparities. From those figures, we see that the disparities are a fairly good nonlinear approximation of the pairwise-cosine similarities. Hence, we can plot the configuration of points that we retrieved by applying MDS on a 2-dimensional space expecting that this representation reflects the original similarities at a satisfactory level (see Figures 7.2(a) and 7.2(b)). For both time spans (1 day and 10 minutes), we can see that spatial proximities do not seem to influence the clusters of locations; there exist some nearby locations which are clustered together



Nonmetric MDS for spatial similarities in Twitter content (1 day)

(a) Document's time span: 1 day

Nonmetric MDS for spatial similarities in Twitter content (10 minutes)

(b) Document's time span: 10 minutes

Figure 7.3: The considered 54 locations of the UK plotted on a 2-dimensional space after applying nonmetric MDS on their pairwise cosine similarities.

but at the same time very distant locations might be members of the same cluster as well. An additional observation visualised in both figures is that the major UK cities (such as London, Manchester, and Liverpool) are situated on the outer space of the 2-dimensional map, whereas locations with smaller population figures have been gathered at centre of this map. In the next section, we go a step further, trying to understand how exactly content is shared among those locations.

7.1.3 Forming networks of content similarity

Based on the previous experimental results, which revealed that proximity does not have a significant influence on content similarity within a country, we are taking a step further concentrating our efforts on explaining spatiotemporal content relations. In particular, we want to examine how Twitter content is shared among different urban centres by proposing a preliminary method for forming networks, which try to capture and describe those relationships.

For a pre-specified set of time intervals $\mathcal{T} = \{t_j\}$, $j \in \{1, ..., \text{m}\}$, we compute the average cosine similarity of all location pairs $\{u_i, u_k\} \in \mathcal{L}_p$ in the same manner as it has already been described in the previous section. $\mathcal{T}$'s length is an arbitrary choice that defines the overall time period encapsulated in the network.

Every location u_k retrieves an Impact Score (**IS**), a quantity expressing the degree of shared content of u_k during $\mathcal{T}$. IS(u_k,$\mathcal{T}$) is defined as the average cosine similarity over all the location pairs in which u_k participates and is computed using:

$$\text{IS}(u_k) = \frac{1}{m} \sum_{\substack{i=1,\, i \neq k, \\ \{u_i, u_k\} \in \mathcal{L}_p}}^{|\mathcal{U}|} \text{Average-Similarity}(k, i), \tag{7.3}$$

where $m \leq |\mathcal{U}|$ is the number of times u_k occurs in a location pair of $\mathcal{L}_p$.

The average pairwise cosine similarities Average-Similarity(i, k) for all location pairs $\{u_i, u_k\}$ $\in \mathcal{L}_p$ are ranked in a decreasing order and the α top ones are being selected. Those α location pairs will form the edges of the inferred spatiotemporal network. The directionality of each edge is decided by the impact scores of the participants: if IS(u_i) $>$ IS(u_j), then

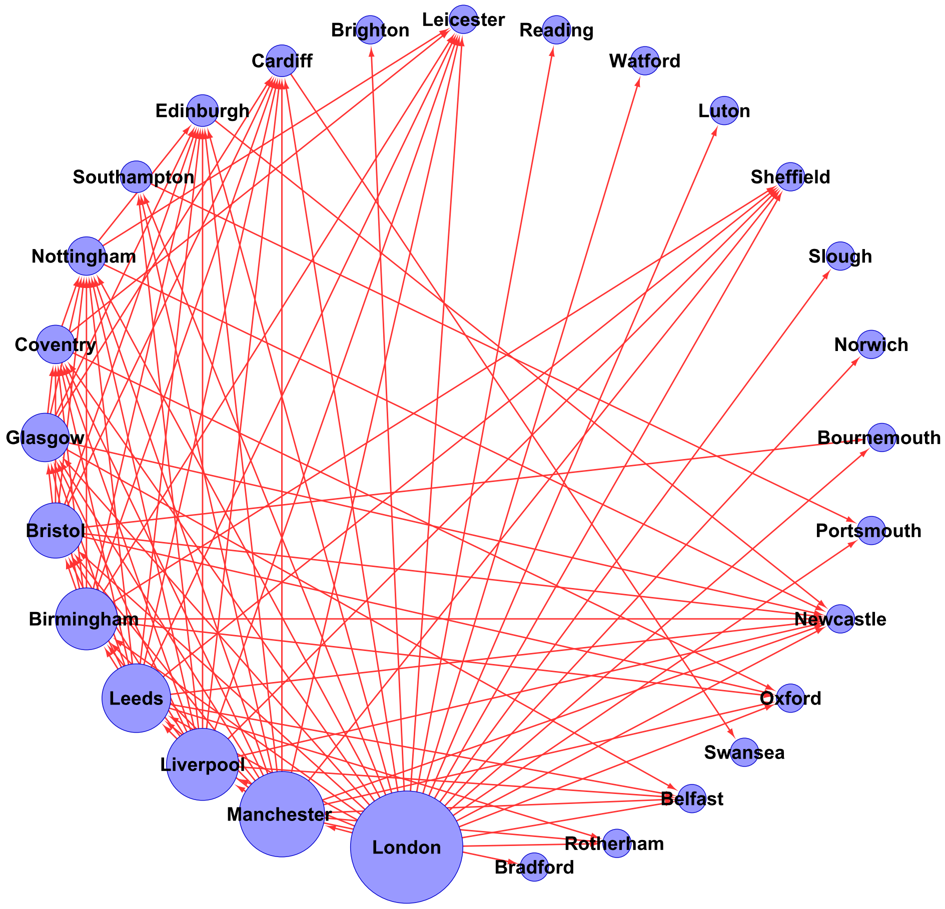


Figure 7.4: Content similarity network for $\mathcal{T}$ equal to 6 months (entire data set) and t_j equal to 1 day. The size of each node is proportional to its out-degree.

.

$u_i \longrightarrow u_j$, else $u_i \longleftarrow u_j$. The numerical value of α is an arbitrary choice; in our experiments $\alpha = 100$ and therefore we expect to infer networks comprised by 100 edges.

First, we set $\mathcal{T}$ equal to the entire 6-month time span of the data set and t_j equal to one day. This will give us an average picture of the network over a 6-month period based on one-day long content similarities between the considered locations. The result is depicted in Figures 7.4 and 7.5.

The network in Figure 7.4 starts from the node with the highest out-degree (London) and then – in a clockwise manner – lists the other nodes in decreasing out-degree order. London is shown to be the most central location in the topic space, followed by Manchester and

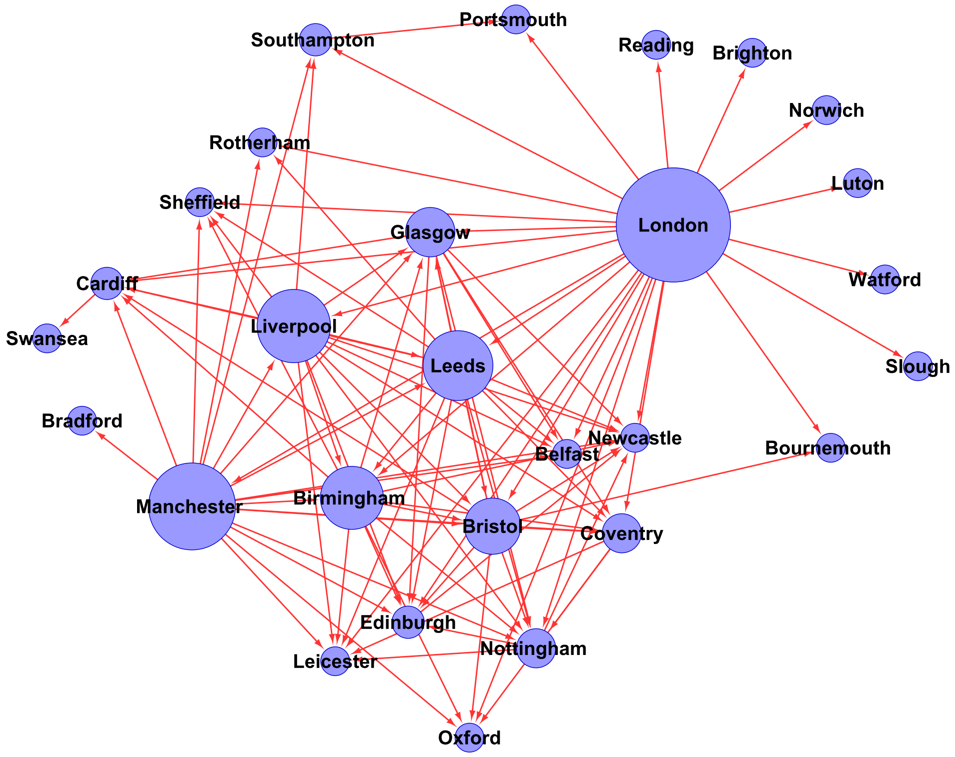


Figure 7.5: The graph of Figure 7.4 in a spring-embedded layout.

.

Liverpool. It is apparent that the locations with the highest out-degrees are the ones with the highest populations; in general, this was something expected to happen on average during a day. Figure 7.5 depicts the same network by using a spring-embedded layout that makes the edges of the network more visible; we can directly observe that London shares content with nearby cities such as Luton and Reading as well as with places that are far away, such as Liverpool and Manchester. Likewise, there is a connection between Cardiff and Swansea, but also between Cardiff and Glasgow.

In the second experiment, we reduce $\mathcal{T}$ to one month (December, 2010) and t_j is set to 10 minutes. Similarly to the previous experiment, Figure 7.6 shows an average picture of the inferred network over December based on 10-minute long content similarities between the considered locations. This network is not directly comparable with the previous one, since it encapsulates data from December 2010 only and not the entire 6-month period; still, we can spot major differences, such as the fact that Manchester is now the most central location in the topic space, followed by Glasgow and London.

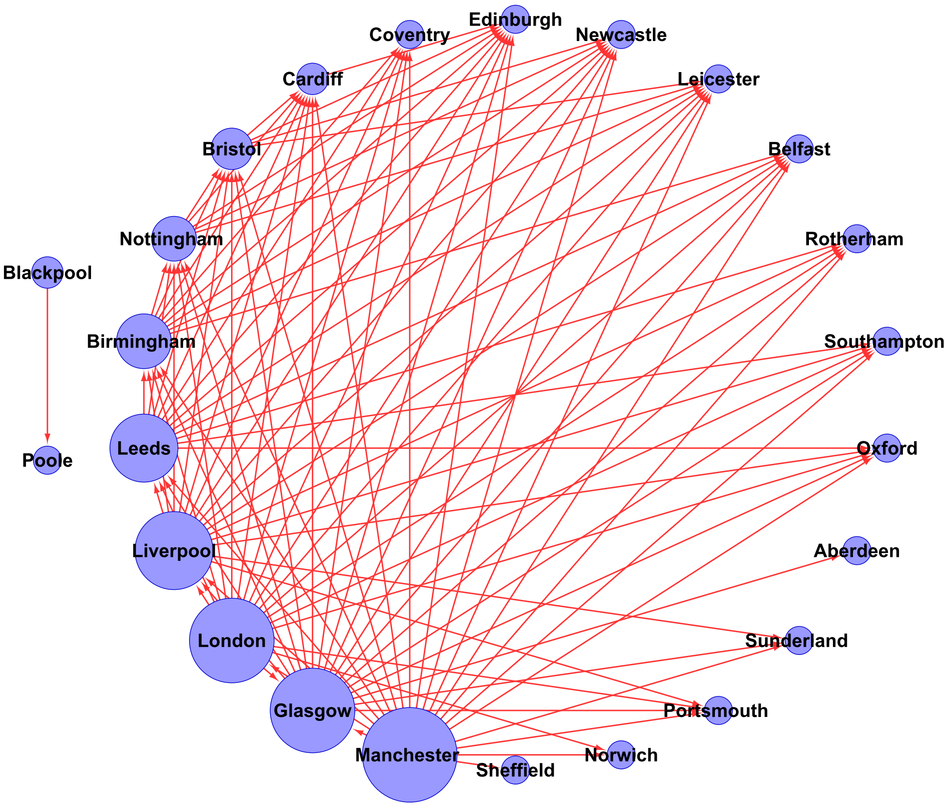
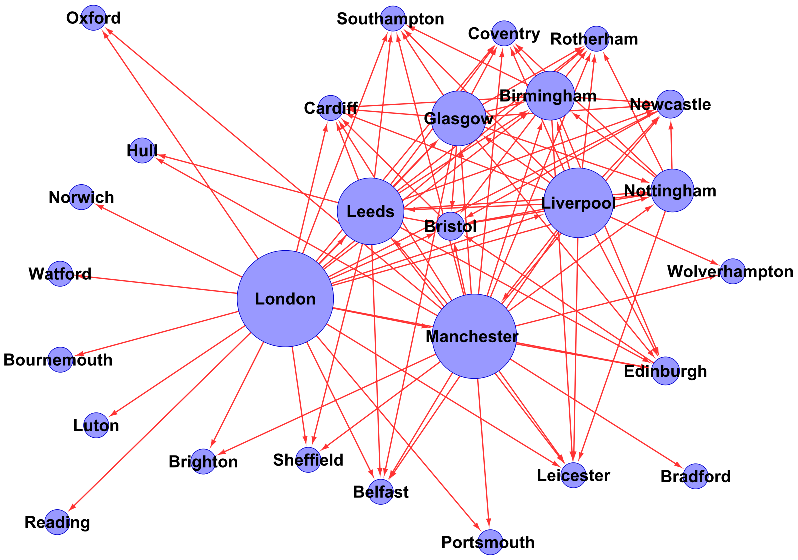


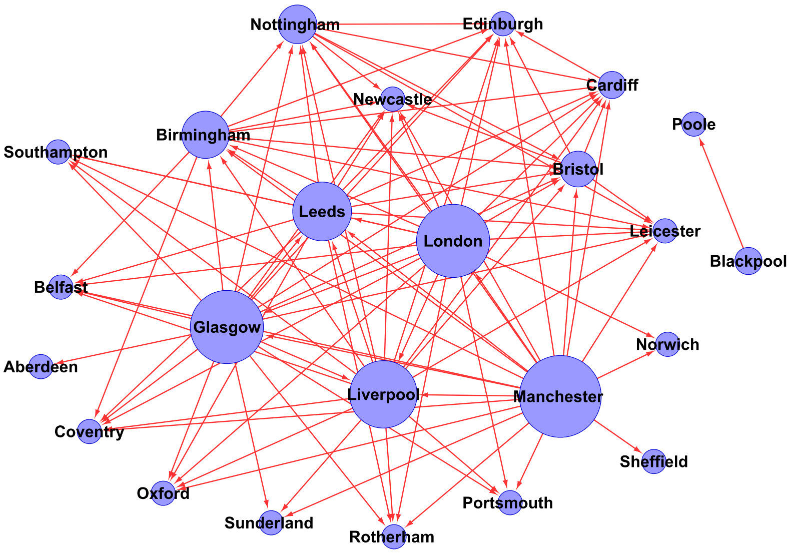
Figure 7.6: Content similarity network for $\mathcal{T}$ equal to 1 month (December, 2010) and t_j equal to 10 minutes. The size of each node is proportional to its out-degree.

Figures 7.7(a) and 7.7(b) show two comparable networks. Figure 7.7(a) is a reduced version of the network in the first experiment, where we have set $\mathcal{T}$ to one month (December, 2010) and t_j to one day, whereas Figure 7.7(b) is identical to the network inferred in the second experiment, but this time displayed using a spring-embedded layout.

The inferred networks under different values of t_j are quite dissimilar. By examining them more closely, we can observe that the one based on short-term (*i.e.* 10-minute) content similarities between locations not only has a different major 'player' (Manchester), but also distributes the out-degrees in a more balanced manner among locations; London's nearby locations are not included in this network and we also see that Glasgow has a significantly increased level of shared content. Additionally, it also reveals a 'strange' connection between Poole and Blackpool – a connection that does not appear in the former network, where t_j was set to 1 day. By applying a network similarity scoring function (introduced in the next section and ranging from 0 to 1 – see Equation 7.5) on those two networks, their similarity



(a) Content similarity network for $\mathcal{T}$ equal to 1 month (December, 2010) and t_j equal to 1 day



(b) Content similarity network for $\mathcal{T}$ equal to 1 month (December, 2010) and t_j equal to 10 minutes

Figure 7.7: Comparing networks encapsulating the same month but using different content similarity time spans.

score is equal to 0.5152; this is a numerical proof on the observable dissimilarity of the two inferred networks.

7.1.4 Network stability

In the previous section, we presented a method for forming content similarity networks among locations over two different time intervals. In this section, we show that the inferred networks are stable over time and therefore, good and well defined descriptions of how different locations share content over time on Twitter. A similar approach to the one introduced in [45] is followed; to prove network stability over time we first form a time series of networks, then apply a similarity metric to compare consecutive networks in this time series, and finally – using a null hypothesis – show that those similarities are statistically significant.

Given that the maximum number of nodes in our networks is known (and is equal to the total number of urban centres considered $|\mathcal{U}|$), we are using a simple similarity measure that compares the directed edges of the networks, known as the Jaccard Distance (**JD**). For two sets of edges $\mathcal{E}_a$ and $\mathcal{E}_b$, JD is defined as:

$$\text{JD}(\mathcal{E}_a, \mathcal{E}_b) = \frac{|\mathcal{E}_a \bigcup \mathcal{E}_b| - |\mathcal{E}_a \bigcap \mathcal{E}_b|}{|\mathcal{E}_a \bigcup \mathcal{E}_b|}. \tag{7.4}$$

JD ranges in [0,1] and is a numerical indication of the dissimilarity between $\mathcal{E}_a$ and $\mathcal{E}_b$. By reversing this notion, we derive the Similarity Score (**SS**) that is used in our experiments:

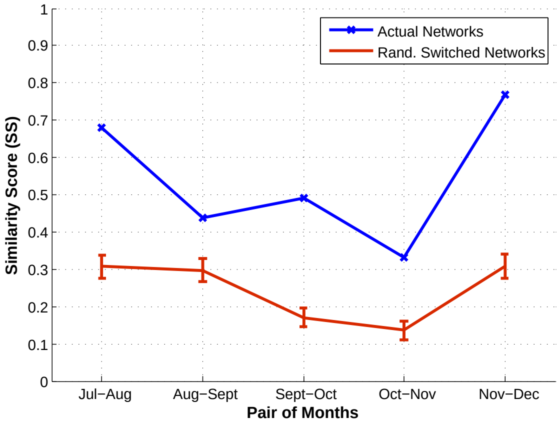
$$\text{SS}(\mathcal{E}_a, \mathcal{E}_b) = 1 - \text{JD}(\mathcal{E}_a, \mathcal{E}_b) = \frac{\mathcal{E}_a \bigcap \mathcal{E}_b}{\mathcal{E}_a \bigcup \mathcal{E}_b}. \tag{7.5}$$

SS ranges also in [0,1], but its value is proportional to the similarity of the networks under comparison.

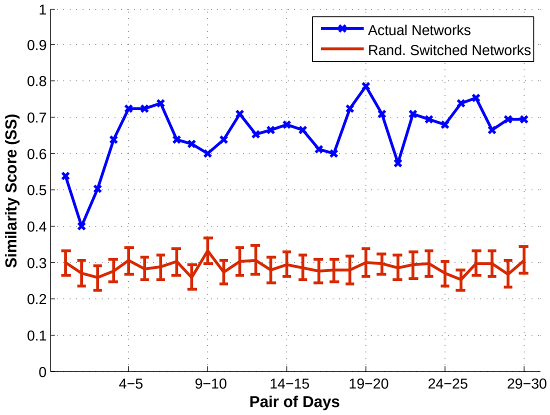
Following similar approaches presented in [45, 112], we base our null hypothesis model on a randomisation strategy able to generate networks with the same degree distribution as the original ones, but randomised topology. This randomisation strategy starts from a given graph and randomly switches edges n times. For example, if edges $u_v \rightarrow u_x$ and $u_y \rightarrow u_z$ are present in the original network, they will be replaced by $u_v \rightarrow u_z$ and $u_x \rightarrow u_y$. In our experiments, when the similarity between two consecutive in time versions of the inferred network – say $\mathcal{E}_{t_1}$ and $\mathcal{E}_{t_2}$ – is tested for statistical significance, randomised switching is performed on $\mathcal{E}_{t_1}$. After $n =$ 1,000 random swaps, the randomised network $\mathcal{E}_{t_1}^r$ is compared to the original $\mathcal{E}_{t_1}$. By counting the times SS$(\mathcal{E}_{t_1}, \mathcal{E}_{t_1}^r)$ is higher than or equal to SS$(\mathcal{E}_{t_1}, \mathcal{E}_{t_2})$, we compute the p-value of the null hypothesis (number of successes divided by the total number of random swaps). P-values lower than or equal to 0.05 justify statistical significance and hence, network stability.

We study the stability of the network over two cases. In the first one, we set $\mathcal{T}$ equal to one month and t_j to one day. In this sense, we are examining whether networks based on one day long content similarities are stable on a monthly basis. In the second, by setting $\mathcal{T}$ to one month (December, 2010) and t_j to 10 minutes, we are investigating whether networks based on 10-minute long content similarities are stable on a day-to-day basis. The results have been plotted in Figures 7.8(a) and 7.8(b) respectively. From those figures, it becomes apparent that the networks are stable on a monthly basis in the first case and on a daily basis in the second, as the similarity score between their consecutive instances is always higher than the one derived by the randomised switching technique (with p-values always smaller than 0.001). As it might have been expected by common logic, the network fluctuates more on a monthly basis, and thus the similarity scores for the first case are lower than the ones observed in the second.

At last, using the data of the second experiment, we investigate how the inferred networks evolve on a 10 minute basis. Following the notation of this section, this is achieved by setting both $\mathcal{T}$ and t_j to 10 minutes. Figure 7.9 shows the derived result. It is a time series of similarity scores drawn on 7-day period (second week of December 2010). The mean similarity score is equal to 0.334 with a standard deviation of 0.11; it is lower – as expected – compared to ones derived by comparing networks of larger time spans. After smoothing with a 24-moving point average, so to extract a 4 hour trend (24 points × 10 minutes per point), an interesting pattern is derived. Similarity scores reach a minimum every day after midnight and during the long night hours and then rise back again, indicating a possible periodic bias. Of course, this could be due to the reduced use of Twitter during the night by individual users but most importantly by professional news and media agencies. This latter observation served as a motivation for the next section, where we study posting time patterns of users on Twitter.



(a) $\mathcal{T}$'s length is equal to a month and time intervals t_j are one hour long



(b) $\mathcal{T}$'s length is equal to one day and time intervals t_j are 10 minutes long

Figure 7.8: Stability of content similarity networks derived by measuring the network pairwise similarity score over consecutive months (a) and days (b).

7.2 Analysing posting time patterns on Social Media

In this section, we are investigating posting time patterns on Twitter. For this purpose, we use approx. 71 million tweets posted from July to December, 2010 and geolocated in the UK. Initially, we show the relative volume (percentage) of tweets per hour by aggregating all days in the data set (see Figure 7.10). The three most distinctive time intervals in terms of posting volume are 12–1p.m. (lunch time), 6–7p.m. (end of working hours for the majority of occupations) and 9–10p.m., which is the hour with the highest number of tweets. We

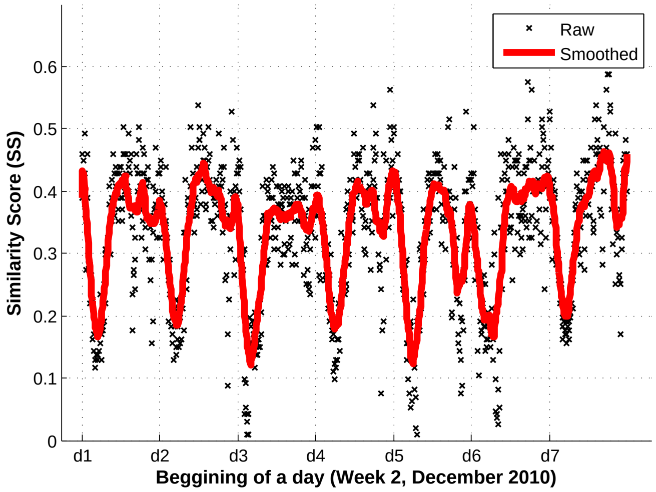


Figure 7.9: Similarity Score time series for consecutive networks with $\mathcal{T}$ and t_j equal to 10 minutes. The red line is a smoothed version of the original plot that uses a 24-point moving average.

also see that Twitter volume goes significantly down during the late night hours and that Twitter is becoming 'alive' again at approx. 7a.m. Therefore, this simple statistical analysis confirms common sense and indicates that Twitter activity might be highly correlated with people's real life activities.

We test the stability of this pattern ($\bar{V}$) by performing the following basic statistical significance test. For every day d_i in our data set we compute the percentages of Twitter volume for each hourly interval (V_{d_i}) and obtain their linear correlation coefficient $\rho(\bar{V}, V_{d_i})$ with the general volume pattern. We also compute the linear correlation between $\bar{V}$ and 1,000 randomly permuted versions of V_{d_i} and count how many times – say N – the correlation of a pseudo-randomised pair was higher or equal to $\rho(\bar{V}, V_{d_i})$. We perform this operation for all 184 days in our data set; hence, the p-value is equal to $N/(1{,}000\times184)$. A p-value $\leq$ 0.05 justifies statistical significance and shows that the extracted pattern is stable. The p-value for the posting times pattern presented in the previous paragraph is equal to 0.004 and therefore, we can assume that it is a stable representation of the hourly volume percentages.

Moving on into this basic statistical analysis, we divide our data set into two subsets: one containing the Twitter volume for week days and one for weekends. If Twitter follows real-life patterns, then we expect an altered behaviour in the posting trends during weekends. Indeed, on weekends Twitter volume peaks in a slightly different time interval (8–9p.m.);

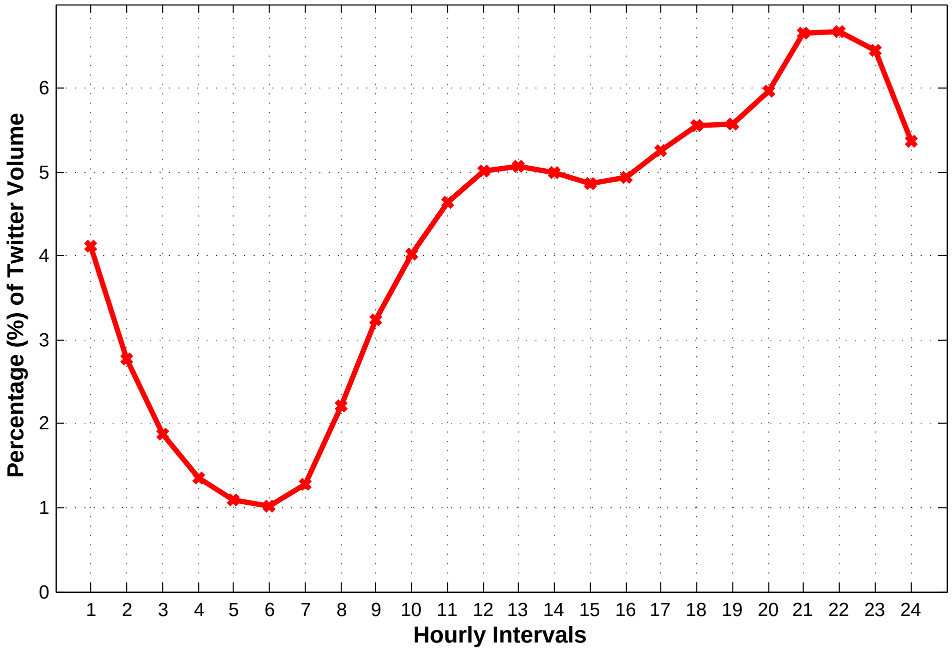


Figure 7.10: Percentage of daily tweets posted per hourly interval based on Twitter content posted between July and December 2010 and geolocated in the UK

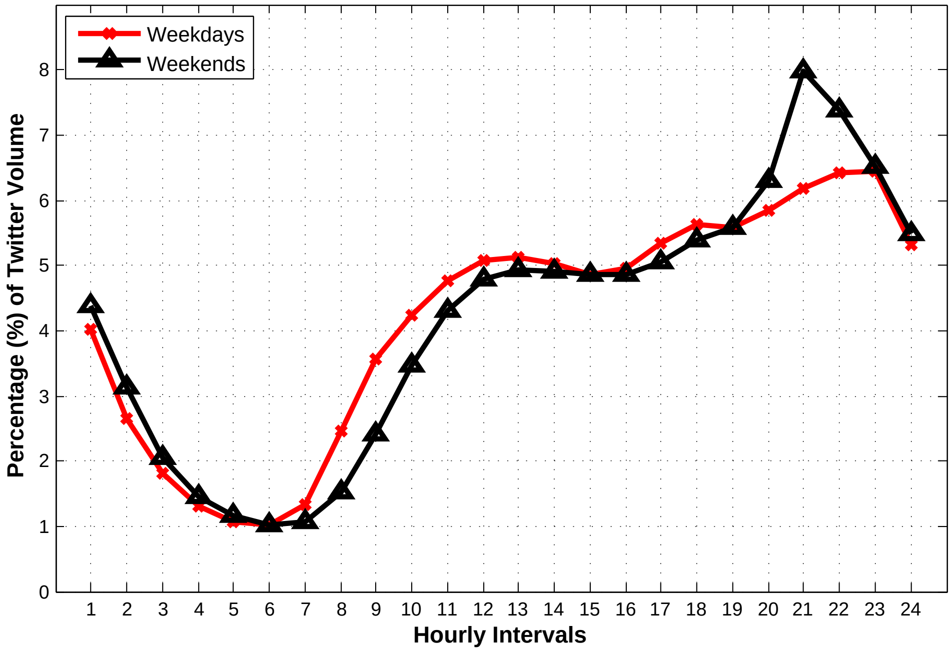
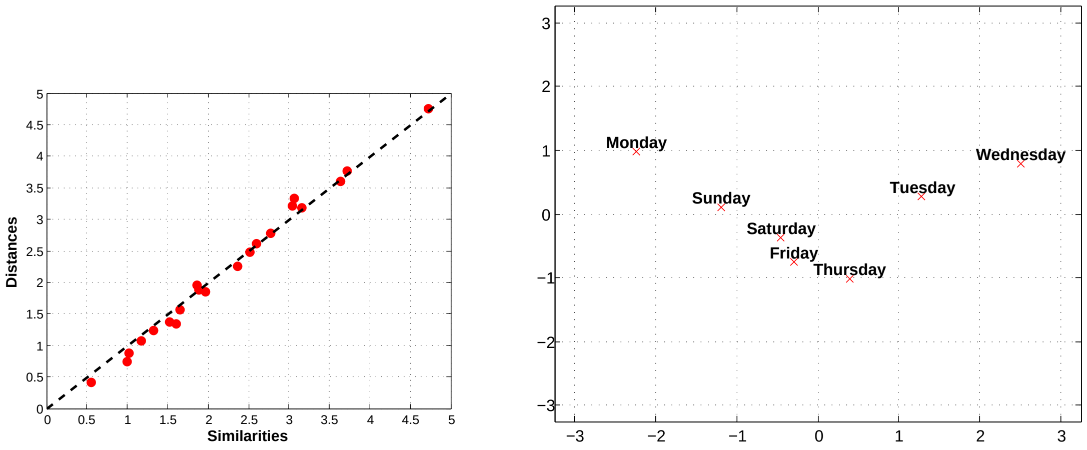


Figure 7.11: Percentage of tweets posted per hourly interval – Week days vs. Weekends



(a) Shepard plot showing goodness of fit for the MDS solution

(b) Clusters of days retrieved by applying MDS

Figure 7.12: Clustering of days by applying Multidimensional Scaling (MDS) based on their posting time patterns.

however, the initial pattern is still there and the two signals (for weekdays and weekends) are very similar – their linear correlation is equal to 0.9594 (p-value: 1.43e-013). More interesting are the deviations occurring during the late night hours showing that Twitter users tend to post more during weekends – possibly because they sleep later than the other days of the week –, as well as the ones in the early morning hours, where a higher volume of tweeting is observed during weekdays. By repeating the previous statistical test, we show that both patterns are stable; the p-values are equal to 0.0052 and 0.001 for weekdays and weekends respectively.

So far it has been shown that there exist at least two slightly different posting time patterns – one for weekdays and another one for weekends. To investigate whether more posting patterns are present and visualise the result, we performed a simple clustering method. Each day is represented by a vector of 24 features, *i.e.* the Twitter volume percentage for each hourly interval. Similarly to Section 7.1.2, we apply MDS; the Shepard plot (Figure 7.12(a)) indicates that the MDS solution presented in Figure 7.12(b) is a good fit. From this figure, we can derive that Monday is the most distinctive day of the week (in terms of its posting volume pattern); we can also observe that consecutive days have been placed in nearby positions of this 2-dimensional space. Interestingly, Friday is more close to Saturday than Thursday and Saturday is more close to Friday than Sunday.

As a general comment, this simple – but also probably interesting – statistical analysis offers some additional proof for the common assumption that Social Media might follow patterns emerging in real life; since the web is an important tool for the majority of people, it is perhaps not easy to differentiate it from their real-life trends.

7.3 A preliminary method for inferring voting intentions from Twitter content

In this section, we present a preliminary method for extracting voting intention figures from Twitter. The case study used to verify our findings is the 2010 General Election in the United Kingdom (UK).[3] In the recent past, a few papers have been published on this topic [54, 102, 109, 120, 164], offering preliminary solutions or discussing the limitations that several approaches might have; we refer to and discuss them at the end of this section.

We consider only the three major parties in the UK; namely the Conservative Party (**CON**), the Labour Party (**LAB**) and the Liberal Democrat Party (**LIBDEM** or **LIB**). Overall, we are using three techniques for extracting positive and negative sentiment from tweets and then two different methods to map this sentiment to voting intention percentages.

Ground truth is acquired by YouGov's Published Results[4] and consists of 68 voting intention polls dated from January to May 2010. Polls usually refer to a pair of days (in our data set only 3 of the them are 1-day polls) and indicate an expectation for the voting intention percentage per political party. As we move closer to the election day (6th of May), they become more dense; there is a new poll published every day. Tweets are drawn from the same period of time, *i.e.* January to May; their total number is greater than 50 million, but not all of them are used as it will become apparent in the following sections. After applying some filtering to keep tweets regarding politics, we end up with 300,000 tweets, *i.e.* approximately 100,000 tweets per political party.

[3] United Kingdom General Election 2010, http://en.wikipedia.org/wiki/United_Kingdom_general_election,_2010.

[4] YouGov Archive, http://labs.yougov.co.uk/publicopinion/archive/.

7.3.1 Extracting positive and negative sentiment from tweets

The common characteristic in all three approaches is that at first we retrieve a set of tweets regarding each political party by using a handmade set of keywords (see Appendix B.4). The keywords are different per party, not many (approx. 60 per party) and as simple as possible; we use the names of the top politicians in each party, the name of the party and so on. The search is case sensitive when the keyword is an 1-gram and case insensitive when it is an n-gram, with $n > 1$. In the latter case, we are not looking for an exact match either; we are just searching for tweets that contain all 1-grams of the target n-gram. Character '#' in front of an 1-gram denotes that we are searching for a Twitter topic. Since those keywords are based mainly on names or entities, one could argue that they could also be created in an automatic manner or extracted from a repository and hence, the human involvement could become insignificant.

The three approaches build on each other; each one is a more elaborate version of its precedent. In the first approach, we are using a stemmed version of SentiWordNet 3.0 to extract positive and negative sentiment from tweets without taking into consideration the different parts of speech (verb, noun, adjective and so on). SentiWordNet is the result of automatically annotating all WordNet synsets according to their degrees of positivity, negativity and neutrality [3, 42]. Stemming is performed by applying Porter's algorithm [129] and Part-Of-Speech (**POS**) tagging is 'skipped' by computing the average positive and negative sentiment weight for each stem over all possible POS that it might appear in. Stems with equal positive and negative sentiment are not considered. The positive and negative sentiment scores of a tweet are retrieved by computing the sum of positive and negative sentiment weights of the words it contains. It is noted that there might exist tweets with no words listed in SentiWordNet; those tweets have zero positive and negative sentiment and therefore are ignored. The acronym **SnPOS** (Sentiment no POS tagging) is used to denote this approach. The motivation behind the removal of POS tagging is the assumption that Twitter language might not follow the norms of formal scripts and therefore POS taggers – trained on more 'mainstream' types of text – might create inaccurate results. However, as we will see in practice in the following sections, this is not always the case, probably because tweets that refer to politics have a much better structure than the more casual ones.

The second approach is based on the same basic principles as SnPOS, only this time POS tagging is applied. When one stem is mapped to a particular POS more than once (this can happen due to stemming), the average positive and negative sentiment weights are assigned to it. POS tagging of tweets is carried out by using Stanford POS Tagger, which is a Java implementation of the log-linear POS taggers described in [162, 163]. By summing over the sentiment weights of a tweet's terms, we retrieve its sentiment score. This method is denoted as **SPOS** (Sentiment with POS tagging).

Finally, we extend SPOS, by incorporating the core word senses of WordNet [111], a semi-automatically compiled list of 5,000 terms.[5] WordNet gives a set of synonyms for each term in this list and we use those synonyms to extend the content of a tweet. That is, if a tweet contains a word listed in WordNet's core terms, the tweet is extended by attaching the synonyms of this word to it. Again, the sentiment score of a tweet is computed by summing over the sentiment weights of its POS tagged terms. This method is identified by the acronym **SPOSW** (Sentiment with POS tagging and WordNet's core terms). The main motivation behind extending the content of a tweet with synonyms is the fact that the short length of a tweet might reduce expressiveness and therefore, adding more words could enhance its semantic orientation. Furthermore, SentiWordNet does not include all English words and by attaching synonyms, we achieve to compute pseudo-sentiment for a greater number of tweets.

7.3.2 Inferring Voting Intentions

By applying a method from the ones described in the previous section, we compute the positive and negative sentiment scores for a set of t tweets, which in turn has been originally extracted by using the set of keywords for a political party. The next task is to turn those scores into a percentage which will represent the voting intention for this particular party. In this section, we describe two methods that address this task.

Optionally, one could first remove tweets with almost equal positive and negative sentiments. The semantic orientation of those tweets is unclear and therefore might not always help. Later on, when we present our experimental findings, for every experimental setting we test, we are also replicating the same experiment by introducing a threshold δ, which removes the top 20,000 (this is equal to approx. 6.67% of the entire number of tweets)

[5] Available at http://wordnet.princeton.edu/wordnet/download/standoff/.

most 'semantically unclear' tweets. Learning the optimal value of δ from data is possible, but also dependent to the existence of a larger sample.

For the remaining $m \leq t$ tweets we compute their mean positive and negative sentiment scores, say μ_{pos_m} and μ_{neg_m} respectively. The sentiment score assigned to this set of tweets is derived by simply subtracting those two quantities

$$\text{Senti}(m) = \mu_{\text{pos}_m} - \mu_{\text{neg}_m}. \tag{7.6}$$

Suppose that we have computed the sentiment scores for a set of n time instances (a time instance is equal to one or two days based on the target voting intention poll) for all three parties, $\text{Senti}^{(n)}_{\text{CON}}$, $\text{Senti}^{(n)}_{\text{LAB}}$ and $\text{Senti}^{(n)}_{\text{LIB}}$. To calibrate the relation between sentiment scores and voting intention percentages, we regress each one of these vectors with the corresponding ground truth using OLS and compute the three weights w_{CON}, w_{LAB} and w_{LIB}. For example,

$$w_{\text{CON}} = \arg\min_{w} \sum_{i}^{n} \left(\text{Poll}^{(i)}_{\text{CON}} - \text{Senti}^{(i)}_{\text{CON}} w\right)^2, \tag{7.7}$$

where $\text{Poll}^{(i)}_{\text{CON}}$ denotes the official voting intention poll percentage for time instance i. We do not introduce bias terms in OLS regression, as during our experimental process it became evident that they receive large values, reducing significantly the freedom of our model and causing overfitting.

In this case study, we are only considering the percentages of the three major parties. After the inference process, the results are normalised in order to represent valid percentages (having a sum equal to 1); in the special (and also rare) case, where inferences are lower than 0, negative results are thresholded to 0 and normalisation does not take place, unless the nonzero percentages sum up to a value greater than 1. Official voting intentions, however, take into consideration the existence of other political 'forces' as well as people's desire not to vote and so on. To create an equivalent and comparable representation, we have normalised the official voting intention percentages as well – based on the fact that the sum of voting intention percentages for the three major parties is on average equal to approx. 90%, this normalisation does not change the picture much (and we can always normalise the results back to a 90% level, if required).

Voting intention inferences are made on unseen data. Again, we have a set of k time instances, which do not overlap with the n time instances used for learning the calibration weights, and we first compute the sentiment scores for each party per time instance. Then, those sentiment scores are multiplied by the corresponding calibration weight; for example the inferred score for the Conservative Party will be equal to:

$$\text{CON}^{(k)} = \text{Senti}_{\text{CON}}^{(k)} \times w_{\text{CON}}. \tag{7.8}$$

After computing the triplet of inferences, $\text{CON}^{(k)}$, $\text{LAB}^{(k)}$ and $\text{LIB}^{(k)}$, we normalise them so that their sum is equal to 1, for example:

$$\text{CON}_{\text{norm}}^{(k)} = \frac{\text{CON}^{(k)}}{\text{CON}^{(k)} + \text{LAB}^{(k)} + \text{LIB}^{(k)}}, \tag{7.9}$$

represents the k normalised inferences of the voting intention for the Conservative Party. We denote the method described above as Mean Thresholded Sentiment (**MTS**).

The second method is identical to MTS, but uses a different function to compute Senti(m) (Equation 7.6). After retrieving the m tweets that satisfy a threshold δ, we count how many of them have a positive sentiment score that is higher than the negative and vice versa. The overall sentiment score of those tweets is then computed by

$$\text{Senti}(m) = \frac{\#\{\text{pos} > \text{neg}\} - \#\{\text{neg} > \text{pos}\}}{m}, \tag{7.10}$$

where $\#\{\text{pos} > \text{neg}\}$ is the number of tweets with a positive sentiment greater than the negative one and $\#\{\text{neg} > \text{pos}\}$ the number of tweets with a negative sentiment greater than the positive. We refer to this method as Dominant Sentiment Class (**DSC**).

7.3.3 Experimental process and results

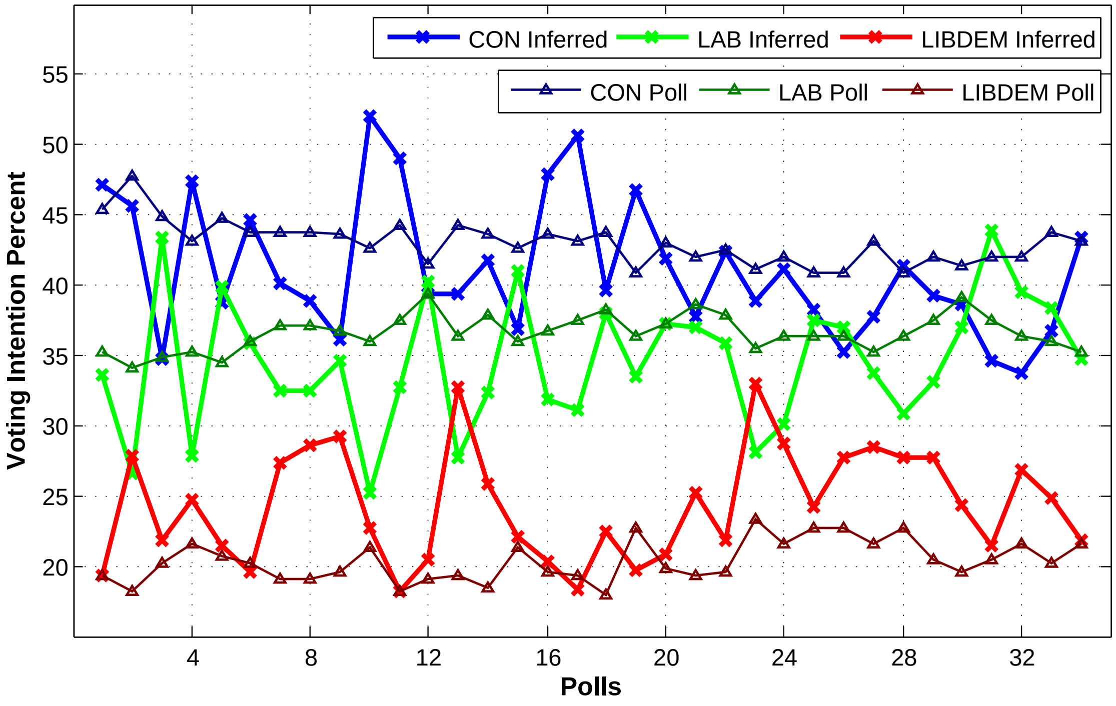
We measure the performance of our methods using two loss functions. Primarily, we use the Mean Absolute Error (**MAE**) (and its standard deviation) between the inferred and the target voting intention percentages. This metric allows for an easier interpretation because it can be read as a percentage, *i.e.* MAE has the same units as the inferred values. In addition, aiming to assess how good is the ranking of the political parties based on the inferred voting intention percentages, we are measuring a Mean Ranking Error (**MRE**). For each triplet of voting intention percentages, the ranking error is defined as the sum over all three parties of the distance between the correct and the inferred ranking. For example, if there exists one incorrect ranking of size 1 (where size measures the difference between a correct and an inferred position), then – since we are dealing with 3 variables only – either there exists one more incorrect ranking of size 1 and therefore the triplet's total error is equal to 2, or two more ranking errors of size 1 and 2, which will make the triplet's total error equal to 4. Obviously, a triplet with correct rankings has an error equal to 0 and the maximum error ($\max(\text{RE})$) per triplet is equal to 4. MRE ranges in [0,1] and is computed by:

$$\text{MRE}(n) = \frac{1}{n \times \max(\text{RE})} \sum_{i=1}^{n} \text{RE}(i). \tag{7.11}$$

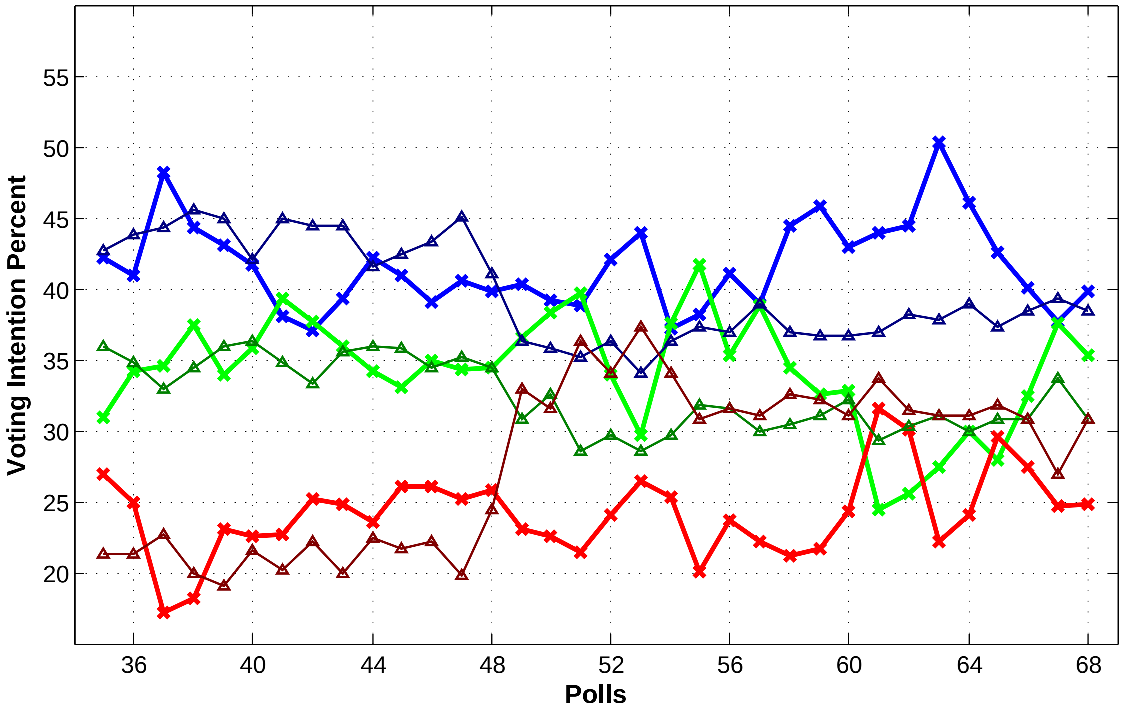
In total, by combining the three methods for extracting positive and negative sentiment from tweets (SnPOS, SPOS and SPOSW) and the two methods for converting sentiment scores to voting intentions (MTS and DSC), we come up with six experimental setups. We run those experiments for $\delta = 0$, *i.e.* for all tweets with a sentiment score, but also for the value of δ which removes the top 20,000 most 'semantically unclear' tweets.

We retrieve the first performance figures by performing leave-one-out cross validation on the aforementioned experimental setups. The performance results for MTS and DSC are presented in Tables 7.1 and 7.2 respectively. Under both methods, we see that thresholding tends – in most occasions – to improve the inference performance in terms of MAE and MRE. SnPOS performs significantly better than SPOS, which also fails to rank voting intentions properly under MTS. Using DCS results in a better performance on average, but does not deliver the overall best performance, which is derived by SPOSW under MTS using a thresholded data set. Figure 7.13 depicts those best performing inferences against the corresponding ground truth; in this case, MAE across all parties is equal to 4.34 with a standard deviation of 2.13 and MRE is 0.1912.

Since our main data set is of small size and based on the fact that the oscillations in the actual voting intention signals are not major (something that might affect the training and testing processes), we also perform a more focused testing. From the 68 polls, we use 1–30 and 47–58 for training and we perform testing on the remaining 26 ones (31–46 and 59–68). The number of tweets that we retrieve using our search terms is increasing as we get nearer to the election day. This is why the training set has been sliced into two parts,



(a) First 34 time instances



(b) Last 34 time instances

Figure 7.13: Leave-one-out cross validation by applying SPOSW and MTS thresholded (in two parts for a better visualisation).

Table 7.1: MAE ± MAE's standard deviation and MRE for Mean Thresholded Sentiment (**MTS**) by performing leave-one-out cross validation. δ denotes the minimum distance between the positive and negative sentiment score for a tweet in order to be considered.

	δ	CON	LAB	LIBDEM	All Parties	MRE
SnPOS	0	12.24 ± 10.66	10.66 ± 10.39	10.1 ± 11.54	11 ± 8.31	0.4559
SnPOS	0.025	12.17 ± 10.63	10.54 ± 10.47	10.11 ± 11.44	10.94 ± 8.27	0.4779
SPOS	0	36.06 ± 7.09	17.86 ± 11.29	15.12 ± 11.65	23.02 ± 5.23	0.9412
SPOS	0.0072	35.99 ± 7.17	17.88 ± 11.26	15.06 ± 11.65	22.98 ± 5.3	0.9412
SPOSW	0	4.22 ± 2.78	3.75 ± 2.86	5.1 ± 3.76	4.36 ± 2.13	0.2059
SPOSW	0.0238	4.17 ± 2.88	3.76 ± 2.82	5.07 ± 3.7	4.34 ± 2.13	0.1912

Table 7.2: MAE ± MAE's standard deviation and MRE for Dominant Class Sentiment (**DCS**) by performing leave-one-out cross validation.

	δ	CON	LAB	LIBDEM	All Parties	MRE
SnPOS	0	9.68 ± 8.82	9.26 ± 8.76	9.34 ± 9.23	9.42 ± 6.66	0.4338
SnPOS	0.025	8.23 ± 6.83	8.66 ± 8.34	9.28 ± 9.12	8.72 ± 6.39	0.375
SPOS	0	11.61 ± 11.99	7.29 ± 5.27	7.9 ± 6.06	8.93 ± 6.21	0.5074
SPOS	0.0072	11.62 ± 11.89	7.52 ± 5.34	8.09 ± 6.33	9.08 ± 6.46	0.5074
SPOSW	0	5.5 ± 4.12	4.19 ± 3.35	5.52 ± 4.82	5.07 ± 3.04	0.2647
SPOSW	0.0238	5.47 ± 3.9	3.95 ± 3.34	5.49 ± 4.49	4.97 ± 2.88	0.2279

Table 7.3: MAE ± MAE's standard deviation and MRE for Mean Thresholded Sentiment (**MTS**). δ denotes the minimum distance between the positive and negative sentiment score for a tweet in order to be considered.

	δ	CON	LAB	LIBDEM	All Parties	MRE	p-value
SnPOS	0	12.97 ± 10.89	10.56 ± 10.05	6.66 ± 7.08	10.06 ± 9.72	0.4038	0.222
SnPOS	0.025	12.98 ± 10.92	10.51 ± 10.03	6.7 ± 7.19	10.06 ± 9.74	0.4038	0.259
SPOS	0	35.03 ± 6.96	21.31 ± 9.4	12.71 ± 7.16	23.02 ± 12.11	0.9231	0.531
SPOS	0.0072	35.14 ± 6.82	21.33 ± 9.41	12.77 ± 7.13	23.08 ± 12.1	0.9231	0.527
SPOSW	0	4.44 ± 3.18	2.66 ± 1.85	3.74 ± 2.86	3.61 ± 2.76	0.1731	0.019
SPOSW	0.0238	4.44 ± 3.31	2.65 ± 1.8	3.84 ± 2.71	3.64 ± 2.75	0.1346	0.006

one 'way' before the election day (05/01 to 24/03) and another much closer (13/04 to 25/04). In this experiment, we additionally retrieve a statistical significance indication for our inferences. To do that, we randomly permute the outcomes of MTS or DSC and come up with a randomised training and test set; we repeat this process for 1,000 times and count how many times a model that is based on randomly permuted data delivers a better inference performance than the actual one – this fraction gives the p-value. We consider that a p-value lower than 0.05 indicates statistical significance.

Inference results for MTS and DCS are presented in Tables 7.3 and 7.4 respectively. Again,

Table 7.4: MAE ± MAE's standard deviation and MRE for Dominant Class Sentiment (**DCS**).

	δ	CON	LAB	LIBDEM	All Parties	MRE	p-value
SnPOS	0	10.56 ± 6.75	9.82 ± 9.18	7.69 ± 9.88	9.36 ± 8.68	0.4038	0.467
SnPOS	0.025	9.66 ± 6.89	9.46 ± 9.05	7.25 ± 9.04	8.79 ± 8.35	0.3654	0.529
SPOS	0	10.63 ± 8.94	8.09 ± 6.37	6.12 ± 5.12	8.28 ± 7.14	0.3846	0.238
SPOS	0.0072	10.51 ± 9.14	7.95 ± 6.18	6.08 ± 5.5	8.18 ± 7.26	0.4038	0.149
SPOSW	0	4.51 ± 3.45	2.87 ± 2.06	3.53 ± 3.29	3.64 ± 3.04	0.1154	0
SPOSW	0.0238	4.49 ± 3.49	2.46 ± 1.81	3.51 ± 3.14	3.49 ± 2.98	0.0962	0

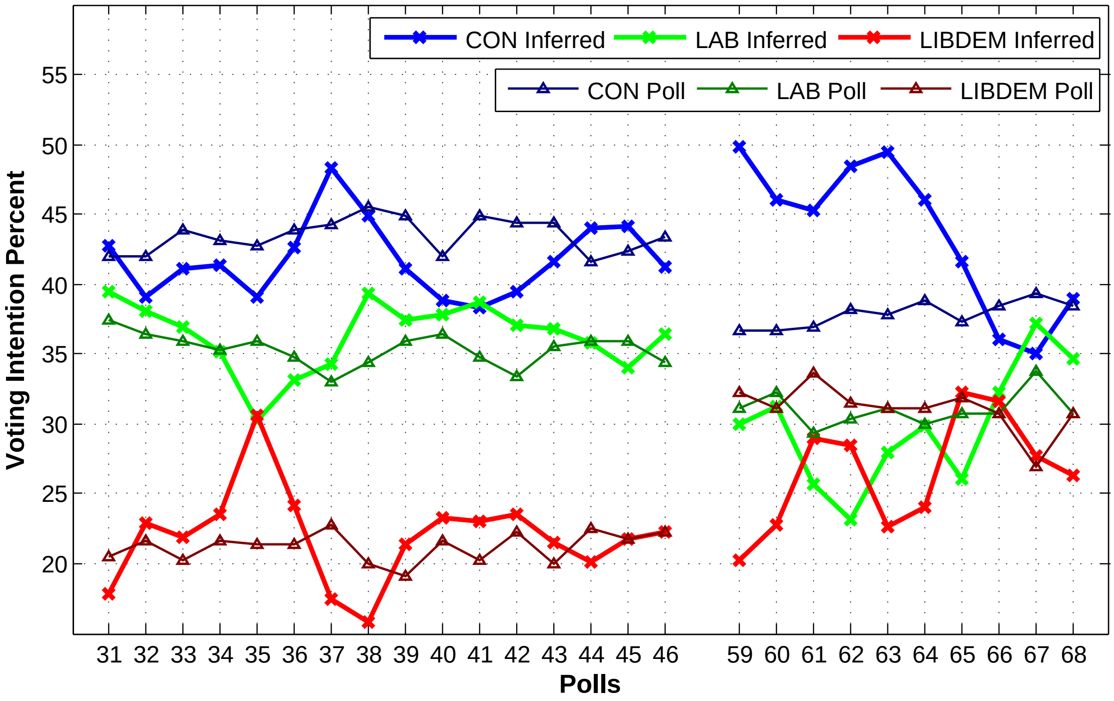


Figure 7.14: SPOSW under thresholded DCS.

it becomes apparent that combining SentiWordNet with POS tagging and extending tweets with WordNet's core senses (SPOSW) gives out the best inference performance as well as that DCS performs on average better than MTS. SnPOS and SPOS not only show a fairly poor performance in terms of MAE and MRE, but also do not deliver statistically significant results. On the contrary, SPOSW's inferences are shown to be statistical significant; its best performance is now achieved under DCS by performing thresholding reaching an MAE of 3.49 ± 2.98 and an MRE of 0.0962 (see Figure 7.14).

7.3.4 Related work and further discussion of the results

The topic of voting intention or electoral result inference from the content of Twitter is quite new in the scientific literature. Tumasjan *et al.* have published a paper, which after providing proof that Twitter is a platform where political discussions are conducted, proposes a method for predicting the result of the German elections in 2009 [164]. Their method uses the Linguistic Inquiry and Word Count (LIWC2007) for semantic analysis [126]; this tool produces a set of 12 dimensions – not just positive or negative sentiment – and therefore in their method they have to introduce an averaging technique in order to acquire 1-dimensional representations. Their method like ours uses sets of keywords (party and politician names) to select politically oriented tweets and then proposes a model for matching the Twitter traffic per party to the final result of the elections. Another paper [120] presents a different method for tracking voting intention polls based on the ratio of positive versus negative sentiment per tweet; their learning approach has some similarities to ours (it is based on linear regression), but only deals with bivariate problems, *i.e.* polls with two outcomes. Nevertheless, a paper published after indicated that the methods in [164] and [120] were problem specific and not very generic, as their application was proven to not do better than chance in predicting the result of the US congressional elections in 2009 [54]. Another paper showed that the popular tool of Google Trends has a limited capacity in predicting election results [102].

Moreover, an interesting paper conducted a further analysis of the aforementioned methodologies and proposed a triplet of necessary standards that any theory aiming to provide a consistent prediction of elections using content from the Social Media should follow [109]. The authors recommend that firstly the prediction theory should be formed as a well defined algorithm, secondly the analysis should be aware of the different characteristics arising in the Social Web and thirdly that the method must include experimental justification on why it works.

Based on those propositions, we tried to formalise the steps in our methodology, we provided statistical significance figures for our results and finally by forming the three different schemes (SnPOS, SPOS and SPOSW) for extracting sentiment from tweets, we tried to encounter some of the special characteristics of Twitter's textual stream. Particularly in SPOSW, where we enriched tweets by adding synonymous words in order to enhance their semantic interpretation, we observed a significant improvement on inference performance.

Similarly to the aforementioned works, the first two sentiment extraction methods that we tried (SnPOS and SPOS) showed poor performance and were not statistically significant. On average, modelling sentiment with DCS performed better than MTS. This is quite interesting as – by definition – DCS compresses more information than MTS; one might argue that this compression results in a much clearer signal, *i.e.* removes some noisy observations, especially when applied on extended tweets. The contribution of thresholding in our experiments is ambiguous; sometimes it improves the inference performance and in most occasions reduces the MRE. As we mentioned in the beginning, the optimal value as well as the contribution of thresholding should be investigated further.

We remind the reader that the presented methods for modelling voting intention polls in this section are preliminary and that several aspects are a matter of future research. Primarily, our methods should be applied to other data sets as well, to come up with experimental proof about their capability to generalise. Another important factor is the choice of keywords that are used to select tweets relevant to the task at hand. We argued that those keywords can be selected from topic-related repositories; still, the influence of each keyword should be modelled. Ideally, an automatic mechanism or algorithm should be compiled in order not only to select an optimal subset of keywords, but also to quantify the contribution of each term. Moreover, possible biases introduced by the use of different sentiment analysis tools should also be considered; tools that incorporate special emotional expressions used in instant messaging and Twitter such as ‘;-)’ or ‘:D’ might achieve a better performance.

A general conclusion that could be extracted or reconfirmed from the presented work is that Social Media do encapsulate content related to public’s political opinion; extending tweets with synonymous terms probably assists in the amplification of this signal. However, it is very important to contemplate that the validity of voting intention polls is questionable. Are those polls a good representation of what is actually happening? Usually, polls from different agencies tend to have significant differences in their percentages and quite often those differences exceed, for example, the lowest total MAE derived in our experimental process (3.49%). Therefore, the formation of a consistently good ground truth is another issue that approaches like this one must resolve.

7.4 Summary of the chapter

In this chapter, we presented some additional but preliminary work trying to exploit further the rich content of Twitter. Firstly, we investigated spatiotemporal relationships based on the textual stream. We showed that content correlation does not necessarily depend on geographical distance within the space of a country. That led to the proposal of a preliminary method capable of forming networks of content similarity among the considered UK locations. Those networks were proven to be stable over time and therefore, a good description of how content is shared across the UK.

Secondly, we focused on extracting posting time patterns on Twitter. The main result showed that Twitter users tend to tweet more as the day evolves with a peak in tweeting between 8 and 9p.m.; as expected, the lowest volume rates occur during the late night hours. More interestingly, this temporal pattern changes during the weekends, where we see that the special characteristics of those days (*e.g.* no work, increased sleeping time in the mornings etc.) can explain the observed deviations from the weekday pattern. By forming clusters of weekdays based on our data set, we also showed that consecutive days are clustered together; the only exception is Monday, which seems to have a more 'independent' behaviour.

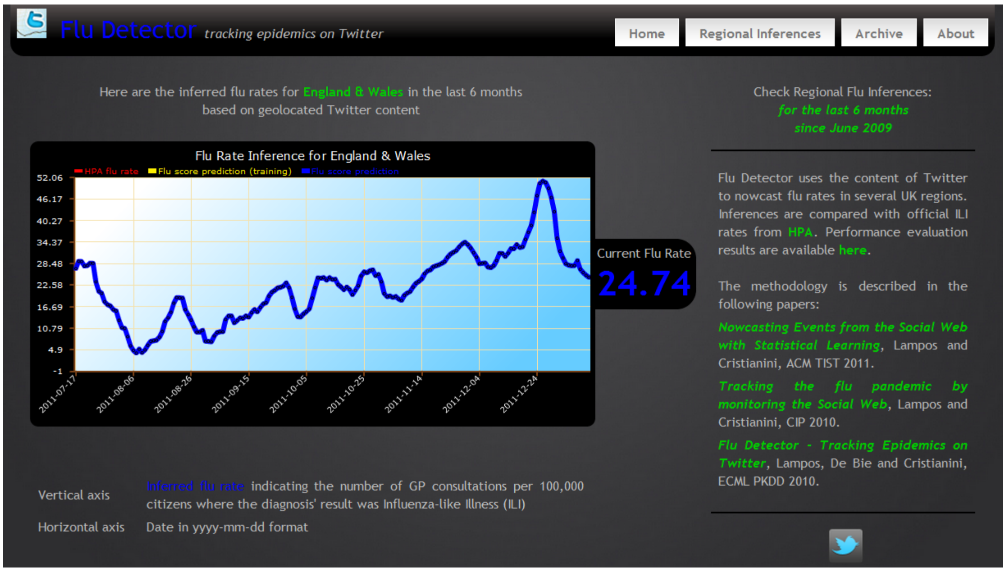
Finally, we presented a preliminary approach for addressing the task of inferring voting intention polls from the content of Twitter, something that could lead to the approximate inference of an election result. Our approach – similarly to other approaches on the topic (*e.g.* [164]) – selects tweets that might refer to politics by using lists of relevant keywords. We have formed three ways for converting those tweets to positive and negative sentiment scores; the first method (SnPOS) applies an averaged version of SentiWordNet [3] disregarding the different parts of speech, the second one (SPOS) considers the different parts of speech by incorporating Stanford POS Tagger [162], and the third one (SPOSW) builds on the second by extending the selected tweets with synonyms from WordNet's core terms [111]. We proposed two ways for converting the sentiment scores of a set of tweets into a voting intention percentage (MTS and DSC); the experimental results indicated that SPOSW (with either MTS or DSC) achieves the best performance.

Chapter 8

Theory in Practice – Applications Driven by our Theoretical Derivations

In this chapter, we present two applications which have been created in our effort to showcase our findings to the academic community and the general public. **Flu Detector** uses the methodology presented in Chapter 5 to infer influeza-like illness rates in several UK regions. **Mood of the Nation** is an application of a mood scoring method (MFMS) presented in Chapter 6 and displays the levels of four types of emotion (anger, fear, joy and sadness) for the same set of UK regions. Parts in this chapter are based on our publication "Flu Detector – Tracking epidemics on Twitter" [93].[1]

[1] Note that since this chapter is short, self-consistent and does not introduce new theory, a summary at the end was considered as redundant and hence, it has been skipped.



Here are the inferred flu rates for England & Wales in the last 6 months based on geolocated Twitter content

Vertical axis Inferred flu rate indicating the number of GP consultations per 100,000 citizens where the diagnosis' result was Influenza-like Illness (ILI)

Horizontal axis Date in yyyy-mm-dd format

Check Regional Flu Inferences:
for the last 6 months
since June 2009

Flu Detector uses the content of Twitter to nowcast flu rates in several UK regions. Inferences are compared with official ILI rates from HPA. Performance evaluation results are available here.

The methodology is described in the following papers:

Nowcasting Events from the Social Web with Statistical Learning, Lampos and Cristianini, ACM TIST 2011.

Tracking the flu pandemic by monitoring the Social Web, Lampos and Cristianini, CIP 2010.

Flu Detector - Tracking Epidemics on Twitter, Lampos, De Bie and Cristianini, ECML PKDD 2010.

Figure 8.1: The front page of Flu Detector's website. The inferred flu scores for the region of England & Wales are displayed.

8.1 Flu Detector

Monitoring the diffusion of an epidemic disease such as seasonal influenza is a very important task. Various methods are deployed by the health sector in order to detect and constrain epidemics, such as counting the consultation rates of GPs [46], school or workforce absenteeism figures [119] and so on. The need of a proper infrastructure and the time delays due to the necessary data processing are the main drawbacks of those methodologies.

We argue that information available on the web can provide an additional means for tackling this problem. In [56, 127] it has been demonstrated that user queries on web search engines can be used to provide an early warning for an epidemic. Furthermore, recent work [2, 92] has shown that the social web media have a predictive power on different domains. In particular, in article [91] we have presented a method for inferring ILI rates for several regions in the UK by using data from Twitter (see also Chapter 5). The core of the method performed feature selection by applying soft Bolasso with consensus threshold validation. In this section, we present a complete pipelined application implementing our theoretical findings titled as the **Flu Detector** (`http://geopatterns.enm.bris.ac.uk/epidemics/` – see Figure 8.1).

Flu Detector is an automated tool with a web interface that is used for tracking the prevalence of ILI in several regions of the UK using the contents of Twitter's microblogging service. The data used to train and validate the models behind this tool are similar to the ones described in Section 5.7. In particular, we use approx. 90 million tweets collected during a period of 392 days (from 21/06/2009 to 25/04/2010 and from 04/10/2010 to 26/12/2010) and geolocated within a 10Km range from the 54 most populated urban centres in the UK. Official ILI rates from the HPA form our ground truth.

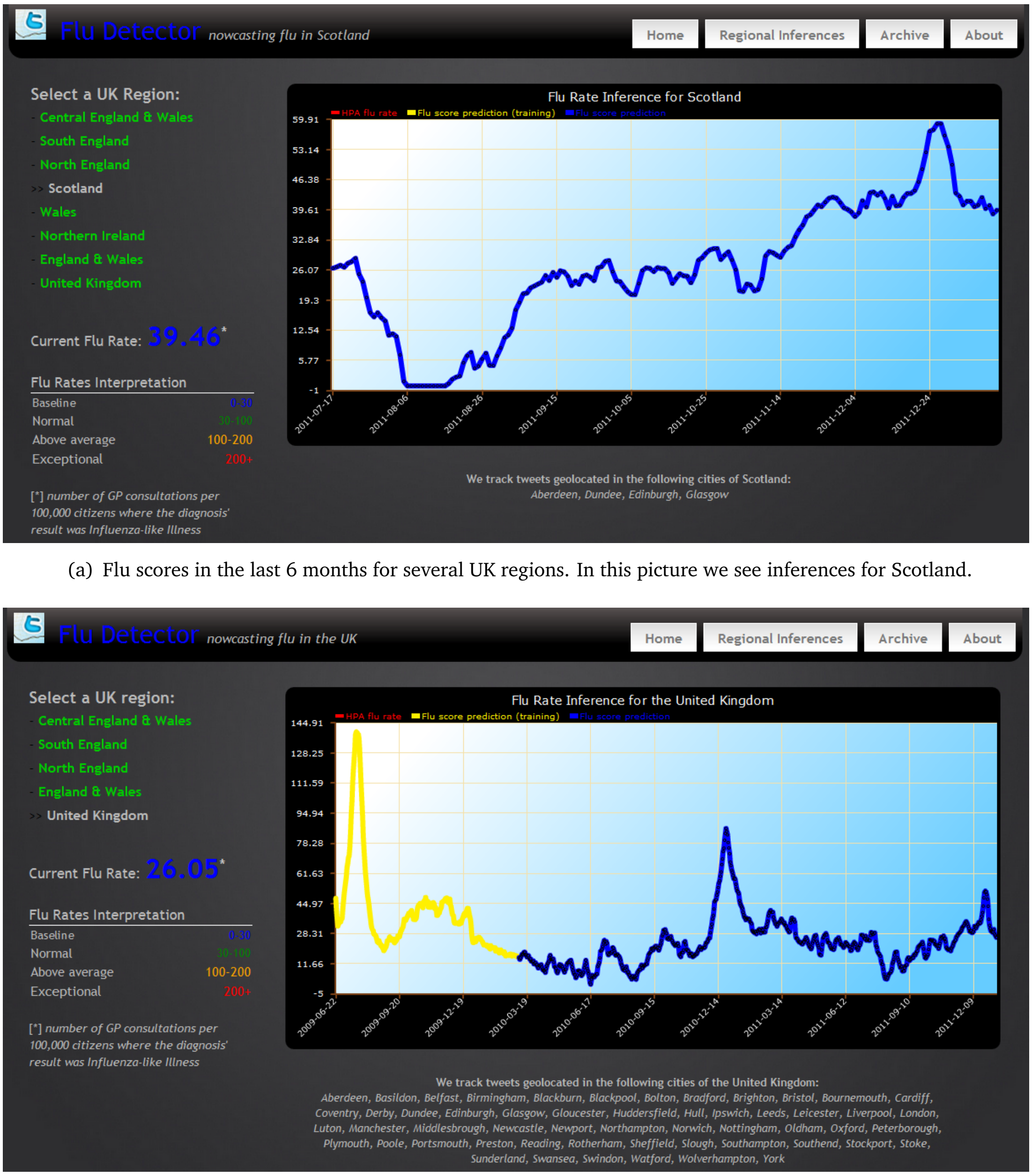
8.1.1 Summary of applied methodology

The applied methodology has been already described in detail in Section 5.4; here we give a short summary of this process. The considered candidate features are formed by 1-grams and 2-grams; the feature extraction process has been described in Section 5.7. Algorithm 4 is applied on the data set comprised by 1-grams as well as on the one of 2-grams and features (1-grams and 2-grams) are selected for all CTs. Then, by applying Algorithm 5 we decide the optimal CT and select a combined set of 1-grams and 2-grams (feature class H). We learn the weight of each n-gram by applying OLS regression. Finally, by combining the selected features, their weights and the daily Twitter content geolocated within a UK region, we are able to compute a daily flu score for this region.

Since an inferred flu score can sometimes be a negative number, something which is not a realistic value given that flu rates should always larger than or equal to zero, we threshold negative inferences to zero. Additionally, based on the fact that flu rates show a smooth behaviour and therefore, no sudden peaks are expected, we also smooth each inference with the non smoothed version of the inferences in the past 6 days. In that way, we also maintain a weekly trend in the displayed scores.

8.1.2 Back-end operations and web interface

Section 3.2.2 has already presented the data collection process. In short, we focus our data collection on tweets geolocated within a 10Km radius from the 54 most populated urban centres in the UK and query Twitter's Search API periodically in order to get the most recent tweets per urban centre. The posts are retrieved in Atom format, parsed using the ROME Java API and stored in a MySQL database.



Flu Rates Interpretation	
Baseline	0-30
Normal	30-100
Above average	100-200
Exceptional	200+

[*] *number of GP consultations per 100,000 citizens where the diagnosis' result was Influenza-like Illness*

We track tweets geolocated in the following cities of Scotland:
Aberdeen, Dundee, Edinburgh, Glasgow

(a) Flu scores in the last 6 months for several UK regions. In this picture we see inferences for Scotland.

Flu Rates Interpretation	
Baseline	0-30
Normal	30-100
Above average	100-200
Exceptional	200+

[*] *number of GP consultations per 100,000 citizens where the diagnosis' result was Influenza-like Illness*

We track tweets geolocated in the following cities of the United Kingdom:
Aberdeen, Basildon, Belfast, Birmingham, Blackburn, Blackpool, Bolton, Bradford, Brighton, Bristol, Bournemouth, Cardiff, Coventry, Derby, Dundee, Edinburgh, Glasgow, Gloucester, Huddersfield, Hull, Ipswich, Leeds, Leicester, Liverpool, London, Luton, Manchester, Middlesbrough, Newcastle, Newport, Northampton, Norwich, Nottingham, Oldham, Oxford, Peterborough, Plymouth, Poole, Portsmouth, Preston, Reading, Rotherham, Sheffield, Slough, Southampton, Southend, Stockport, Stoke, Sunderland, Swansea, Swindon, Watford, Wolverhampton, York

(b) Archive of flu scores (since June, 2009) for several UK regions. In this picture we see inferences for the entire UK.

Figure 8.2: Sample pages of Flu Detector's website.

The learning process – described in the previous section – is performed offline. An automated procedure updates the flu-score inferences on a daily basis and uploads them on the website. Apart from the main three regions (North England, South England, Central England & Wales) on which the training procedure was based, we also display flu score projections for Northern Ireland, Scotland, Wales, England & Wales and the entire UK.

Flu Detector's website is running on an Apache server using common web technologies (HTML, CSS and PHP). The charts are Flash applications based on the Open Flash Chart API.[2] Figure 8.2(a) shows the page of the website where inferred flu scores of the last 6 months are displayed for the aforementioned regions (accessible via a navigation menu). The inferred flu rate of the previous day, *i.e.* the current flu score, is also displayed on a separate box on the left. Recall that a flu score denotes the number of ILI-diagnosed patients in a population of a 100,000 citizens; rates between 0 and 30 are considered as baseline, between 30 and 100 as normal, 100 and 200 as above average and if they exceed 200, then they are characterised as exceptional. Finally, Figure 8.2(b) shows the page of the website, where the entire flu rate time series is displayed (since June 2009).

8.2 Mood of the Nation

Social media can be perceived as another way to look into the society [145]. In particular, work has shown that the 'mood' of tweets is able to predict the stock market [11] as well as detect significant events or situations emerging in real-life ([95] and Chapter 6). In this section, we present a complete pipelined application implementing some of our theoretical findings titled as **Mood of the Nation** (http://geopatterns.enm.bris.ac.uk/mood/ – see Figure 8.3(a)).

8.2.1 Summary of applied methodology

The method that we applied in order to compute the displayed mood scores has been already described in Sections 6.1.1.1 and 6.2.1 in detail. Mood of the Nation displays the scores for four emotions, namely anger, fear, joy and sadness, based on the affective terms included in WordNet Affect [153] (see also Appendix B.3). In particular, we have applied

[2] Open Flash Chart API, http://teethgrinder.co.uk/open-flash-chart/.

the Mean Frequency Mood Scoring (MFMS) scheme which is based on the assumption that the importance of each term is proportional to its frequency in the corpus.

We do not display raw mood scores but their z-scores, *i.e.* from each score we subtract the mean and then divide the remainder with the standard deviation of the raw mood score. In that way, all mood scores are comparable to each other and can be plotted in a single figure. A positive score $m \in \mathbb{R}$ indicates that this mood type is m standard deviations larger than its mean value; the opposite interpretation stands for a negative mood score. We have chosen to multiply all standardised mood scores (positive or negative) with a constant number equal to 10, to allow for a better presentation of the final results to the end user. Another choice of ours was to maintain a dynamic mean and standard deviation for each mood type, which means that every day – as new data comes – those values are automatically adapted.[3] As a final note, in contrast with Flu Detector's rates, here smoothing does not take place as it might 'hide' sudden emotional peaks which are of interest.

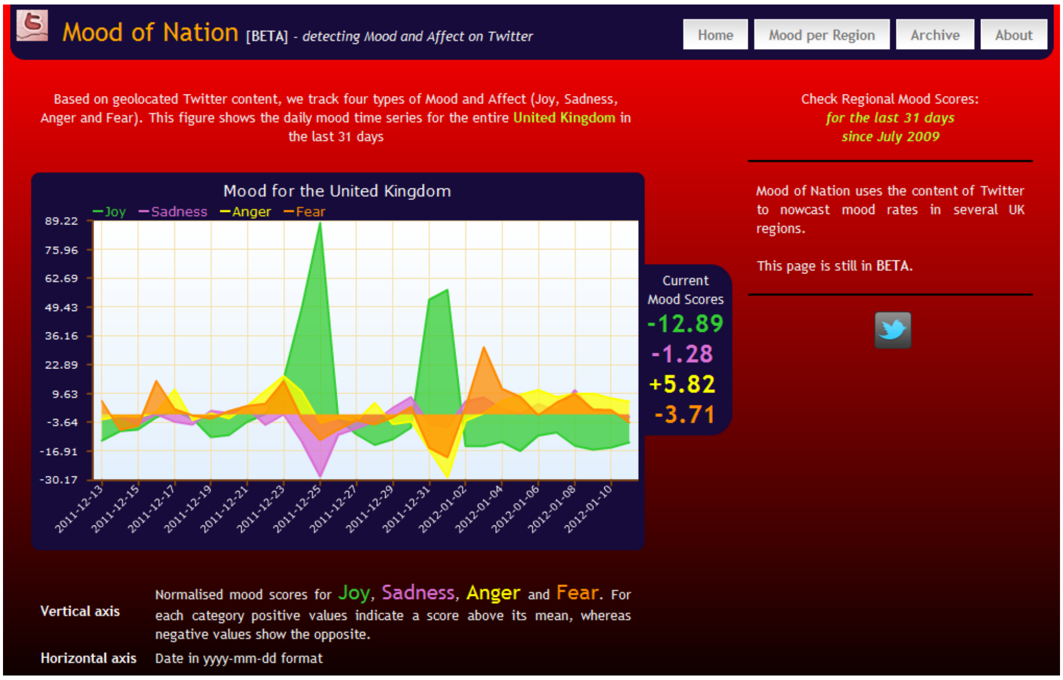
8.2.2 Back-end operations and web interface

Data collection and other back-end operations are carried out in a similar manner as in Flu Detector. We kept the same skeleton for the website with a small amount of changes and also maintained the considered UK regions. Of course, we might be using the same Twitter data set, but now the core computation is different. Therefore, MFMS scheme is implemented in Java and updates the scores for the four mood types on a daily basis.

Every page of Mood of the Nation shows one chart that includes semi-transparent plots of all four mood types (coloured in yellow, orange, green and pink for anger, fear, joy and sadness respectively). Figure 8.3(a) shows the main page of the website where mood scores for the entire UK are displayed, whereas Figure 8.3(b) is the page where the mood scores of the last 30 days are displayed per region. Another interesting page is the one that displays historical mood data (starting from June 2009),[4] where for example one can observe – among more interesting observations discussed in Chapter 6 – peaks of 'joy' during the Christmas day for 2009, 2010 and 2011.

[3] In our future plans is to allow the users to create and try different means directly from the website's interface.

[4] No figure is available for this page, but it can be found at http://geopatterns.enm.bris.ac.uk/mood/twitter-mood-archive.php?region=UK.



Mood of Nation [BETA] - *detecting Mood and Affect on Twitter*

Home | Mood per Region | Archive | About

Based on geolocated Twitter content, we track four types of Mood and Affect (Joy, Sadness, Anger and Fear). This figure shows the daily mood time series for the entire United Kingdom in the last 31 days

Check Regional Mood Scores:
for the last 31 days
since July 2009

Mood of Nation uses the content of Twitter to nowcast mood rates in several UK regions.

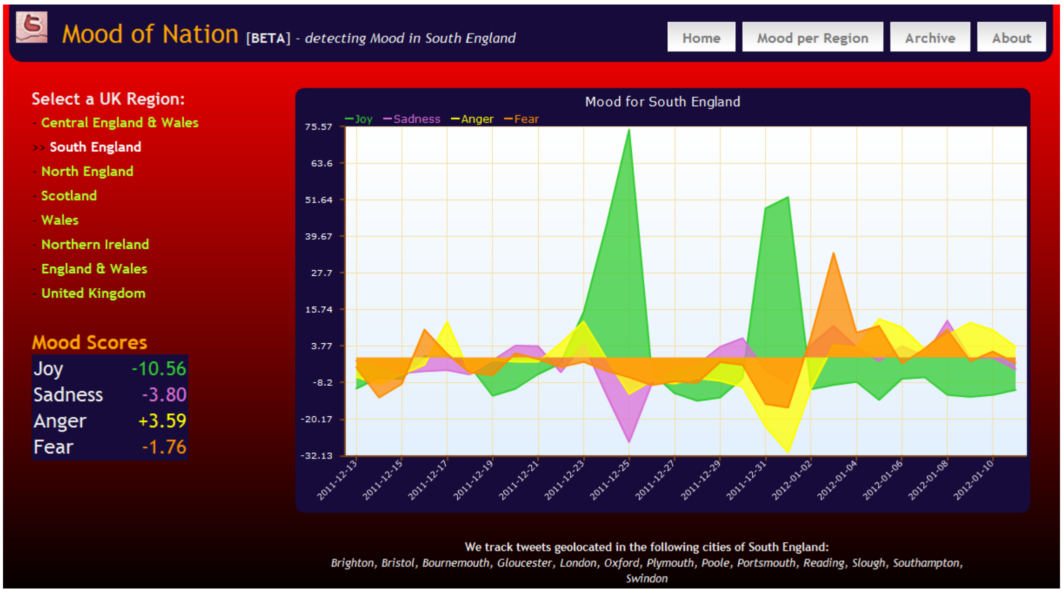
This page is still in BETA.

Current Mood Scores
-12.89
-1.28
+5.82
-3.71

Vertical axis Normalised mood scores for Joy, Sadness, Anger and Fear. For each category positive values indicate a score above its mean, whereas negative values show the opposite.

Horizontal axis Date in yyyy-mm-dd format

(a) Front page of Mood of Nation where the inferred scores for Anger, Fear, Joy and Sadness in the entire UK are displayed.



Mood of Nation [BETA] - *detecting Mood in South England*

Home | Mood per Region | Archive | About

Select a UK Region:
- Central England & Wales
- >> South England
- North England
- Scotland
- Wales
- Northern Ireland
- England & Wales
- United Kingdom

Mood Scores

Joy	-10.56
Sadness	-3.80
Anger	+3.59
Fear	-1.76

We track tweets geolocated in the following cities of South England:
Brighton, Bristol, Bournemouth, Gloucester, London, Oxford, Plymouth, Poole, Portsmouth, Reading, Slough, Southampton, Swindon

(b) Mood scores for the last 30 days for several UK regions. In this picture we see inferences for the region of South England.

Figure 8.3: Sample pages of Mood of the Nation website.

Chapter 9

Conclusions of the Thesis

This chapter summarises and closes the Thesis. We start by providing the reader with a chapter-by-chapter synopsis of the script. Then, we discuss the outcomes and conclusions of this work from a more general perspective, referring also to its impact on the academic community. At the end of this chapter, we propose further research directions that might be interesting to explore in the future.

9.1 Summary of the Thesis with hindsight

In **Chapter 2**, we summarised the theoretical background behind our research. The core scientific field of our work is primarily defined by Statistical Machine Learning, but we also apply methods from the discipline of IR. Amongst other theoretical notions, this chapter also introduced the reader to regularised regression, explained how CARTs operate and introduced various vector space models.

Chapter 3 gave a characterisation of Twitter, the social network that is used as the primary source of information for this project, firstly by describing its main attributes and then by referring to several 'Twitter-driven' studies proving the significant value of this content in practice. In the second part of this chapter, we explained how we crawl, store and then process Twitter data, and also described the information sources that served as ground truth in our experiments.

Chapter 4 presented the first and preliminary method for tracking a flu epidemic or in general the diffusion of ILI in the population by exploiting the content of Twitter [92]. Starting from a manually formed set of illness related keywords, which were proven to have well correlated frequencies with the official flu rates, but then moving on and proposing a fully automated methodology that used a sparse regressor, the LASSO [158], we were able not only to extract textual features (1-grams) that had a close semantic connection with the underlying topic, but also to infer accurately the flu rates for several regions in the UK during the H1N1 epidemic (2009). However, this approach faced some limitations mainly because LASSO is not a model-consistent learner [103, 173], 1-grams did not always encapsulate a valid semantic description of the topic, and no proof was provided for a possible generalisation of the experimental results.

In **Chapter 5**, we presented a more generic methodology for inferring the occurrence and magnitude of an event or phenomenon by exploring the rich amount of unstructured textual information on the social part of the web – in particular, Twitter – [91], by improving on the methods presented in the previous chapter. Linear (bootstrap LASSO with consensus threshold validation combined with OLS regression) and nonlinear (ensemble of CARTs) learning functions have been applied in the core of our methodology and the inference performance has been compared to other techniques used to address similar problems [31, 56]. In this study, we have used two types of features, 1-grams and 2-grams, and have also investigated several hybrid combinations of those feature classes in our effort to nowcast two phenomena emerging in real life: flu and rainfall rates. Whereas the importance and usefulness of inferring flu rates is a clear fact, rainfall rates inference mainly served as a benchmark and much harder problem that tested the capacity of our proposed inference schemes.

In **Chapter 6**, we explored the same data source (*i.e.* Twitter content) for a different problem: the extraction of seasonal circadian [94] and daily mood patterns [95]. Our analysis was based on two methodologies, one (MFMS) that assigns emotion-triggering words with an importance weight analogous to their frequency in the text stream and another one (MSFMS) which standardises term frequencies and therefore removes biases induced by frequent emotional markers. We investigated four mood types, namely anger, fear, joy and sadness. The inferred circadian patterns were partly confirmed by research results from the field of Psychiatry, but contradicted with the ones presented in a similar recent work [57]. In general, PA was shown to be stronger that NA during the daytime and vice versa during the night hours. Looking into daily mood patterns, we gained evidence supporting that real-life events affect the mood of Twitter users; in 2011, for example, the UK riots, Winehouse's death or the earthquake in Japan caused an observable increase in the NA of Twitter users geolocated in the UK. Both circadian and daily mood patterns have some periodical features; the most common period for all mood signals is the length of a week.

Chapter 7 presented three additional Pattern Discovery studies on Twitter content that could serve as basis for future research directions. The first one investigated spatiotemporal relationships in the textual stream. Within the limits of a country, we showed that similarities in Twitter content do not depend on the distance between the locations, where the published content is authored; then, we proposed a preliminary method for forming networks of content similarity among the considered UK locations and demonstrated that the inferred networks are stable over time. Secondly, we focused on extracting posting time patterns on Twitter distinguishing between weekdays and weekends. Interestingly, those patterns seem to confirm common sense as they match with behavioural norms drawn from real-life. Finally, we reported on a preliminary set of methods for addressing the problem of voting intention inference from Twitter content using the 2010 UK General Election as a case study.

Lastly, in **Chapter 8** we presented two online tools, which showcase some of our findings to the academic community and also serve as dynamic applications of our theoretical derivations. 'Flu Detector'[1] detects influenza-like illness for several UK regions [93], whereas 'Mood of the Nation'[2] displays mood scores (anger, fear, joy and sadness) for the same set of UK regions; both tools use Twitter content as their input information.

9.2 General conclusions

The work presented in this Thesis is an additional and consistent proof that information of high value does exist in unstructured, user-generated web text streams. This information reflects on various aspects of real-life already and as the number of people using and conducting parts of their life through the Social Media constantly increases, it is logically expected to reflect on even more.

Throughout this Thesis we demonstrated several ways for extracting types of information from the social web and especially from Twitter's microblogging service. More specifically, we focused on extracting identifiable moments emerging in real-life, commonly referred to as events, as well as more ambiguous trends, which are encapsulated in the general task of Pattern Discovery. Our case studies on flu and rainfall rates inference (or nowcasting) showed that events, which tend to oscillate between active and inactive moments and – at the same time – may be a topic of discussion in the Social Media, could be successfully detected and quantified. Additional work showed that the textual stream encloses affective norms of the general population and delivered common life patterns, such as weekly periodic behaviours or distinctive emotional norms during weekends. Content similarities through time were used to form stable spatiotemporal networks of urban centres which reveal how information is shared among them. By combining statistical learning with NLP, we were in the position to produce a function, which turns Twitter content to voting intention figures and therefore propose a preliminary method for this interesting task.

An important and crucial part of our work was the verification of our methodologies. In a supervised learning scenario, this is usually part of the problem's formulation and standard loss functions or similarity metrics can be applied, based on the fact that a ground truth

[1] Flu Detector, Link: http://geopatterns.enm.bris.ac.uk/epidemics/.

[2] Mood of the Nation, Link: http://geopatterns.enm.bris.ac.uk/mood/.

is available to compare with. In an unsupervised scenario, one has to verify the acquired results with facts – numerical, statistical or more abstract – which might not be directly comparable with them. Based on the performance of our supervised methods for nowcasting events or inferring voting intention polls, we can strongly argue that information on the social web is a good representation of real life. This conclusion is becoming stronger by observing how well the results of mood analysis fit with real-life happenings such as sudden deaths of famous personas, celebrations or public unrest.

We made an effort to develop as generic and generalisable methods or algorithms as possible. On the one side, the entirety of the proposed methodologies is directly applicable to other textual content streams. But most importantly, by investigating distinct case studies, we also acquired experimental proof – both in terms of semantic relativity and measurable distance from a target topic or variable – that those techniques would be able to extract several types of signals; not only the specific ones that we were in the position to test. Our experiments might have been based on one social platform only, but this, of course, should not be considered as a significant limitation as we avoided to use the specific structural features of Twitter.

Wherever 'early' success was achieved,[3] we also tried to extend and improve our methodologies. The use of 2-grams, for example, increased the semantic connection between the automatically selected terms and the target topic, and the hybrid combinations of 1-grams and 2-grams improved further the inference performance. LASSO's model selection inconsistencies have been resolved by bootstrapping and nonlinearities have been explored by using CARTs. Lastly, experimental proof indicated that enriching the content of tweets with synonyms might improve the performance of sentiment analysis on them.

9.3 Work impact on academic research

Our publications were well-received by the academic community and several new papers referred to or built on their basic ideas; in this short paragraph we refer to some of them. The detection of flu or other infectious diseases using the contents of the social streams together with geolocation properties – in some occasions – became a research subject in many conference or journal publications [26, 31, 58, 84, 116, 123, 124, 148] as well as

[3] Early acquires a relative definition which is decided by the strict time limits that a Ph.D. project introduces.

the main theme of a recent book [167] serving the newly introduced topic of 'Infoveillance' [43]. Another research subject with increasing interest is the extraction of voting intentions by applying sentiment analysis on tweets [102, 109]. Finally, the detection of environmental incidents and possible hazards by observing social networks and especially Twitter has been another important topic [34, 151].[4]

9.4 Research directions for future work

This work could be extended in several directions and we are glad to report that many of them are already explored by a growing academic community, something that can be easily proven by the increasing number of EU-funded projects and scientific publications on Social Media analysis.

A research direction is always the one of improving the core theory of the proposed methods, algorithms and techniques. More specifically, one could investigate the application of even more sophisticated linear, nonlinear or generative learners. The experimentation with additional case studies might also give out new information about the key properties that the learning function should hold. In addition, the mixture of linear with nonlinear learning functions could also be of interest. As far as the feature space is concerned and based on the fact that 2-grams showed a much better semantic correlation with the target topics, one could also investigate this further using n-grams with $n > 2$ or a formalised set of regular expressions aimed to look for, identify and match with specific textual characteristics.

Another interesting aspect for future exploration is the incorporation of mood or sentiment analysis in the event detection methodology. It is rational to consider that events emerging in real life affect the mood in Social Media as we already have proof for this. Therefore, this relationship needs to be formalised, so that those affective norms could assist in the detection or quantification of at least a class of events in the effort of detecting or preventing potential social dangers.

Our methods have been concentrated on textual content written in English language only, hence porting our methods to other languages by applying Machine Translation frameworks could be of interest. Going a step further, an exciting challenge would be to successfully

[4] At the moment (April 2012), our work has been cited in more than 40 scientific publications.

fuse different types of information; in this work we only used textual information, but the web offers much more. Information in images, podcasts and videos should be explored in combination with the signals from the textual stream.

Important challenges also emerge in combination with other scientific fields. This vast amount of information provides for example the social scientists or psychiatrists with the opportunity to answer questions, which before have been considered as experimentally infeasible. On the other hand, results derived by a statistical analysis usually need the interpretation of experts from other disciplines. Therefore, interdisciplinary research paths should naturally be explored.

As a final but equally important note, we have to warn the academic community for privacy issues imposed by all those new developments. Those issues should be taken very carefully in consideration, as all the proposed methods, could indirectly expose and stigmatise users and therefore, people. Computer scientists must work closely with legislators to reassure that all this automation will improve the quality of life and will not restrict society's freedom of speech and right to privacy.

Appendix A

Basic Theoretical Notions

A.1 Arithmetic Mean, (Co)Variance, Standard Deviation and Z-score

The arithmetic mean for an N-sized sample of a discrete variable $X = \{x_1, x_2, ..., x_N\}$ is given by:

$$\mu_X = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{A.1}$$

The variance σ_X^2 of the above sample is defined as

$$\sigma_X^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_X)^2, \tag{A.2}$$

whereas the standard deviation σ_X is equal to $\sqrt{\sigma_X^2}$.

The covariance of two variables X and Y of size N is given by:

$$\text{cov}_{XY} = \sum_{i=1}^{N} \frac{(x_i - \mu_X)(y_i - \mu_Y)}{N} \tag{A.3}$$

A zero covariance means that X and Y are uncorrelated; if $\text{cov}_{XY} > 0$, then as X increases, Y tends to increase and if $\text{cov}_{XY} < 0$, then Y decreases, as X increases.

In the above formulas, we are making the assumption that the samples are independent and identically distributed (*i.i.d.* samples).

Z-score is a method used for standardising the samples of a variable, using its sample mean and standard deviation. The numerical result of this standardisation shows how many standard deviations away from the mean is the value of each sample. In our example, x_i will be transformed to:

$$x_i^{(z)} = \frac{x_i - \mu_X}{\sigma_X}, i \in \{1, 2, ..., N\}. \tag{A.4}$$

A.2 Variance and bias of a predictor

Suppose that $y = f(X) + \varepsilon$, where y is the target variable, X denotes a set of observations, $f(\bullet)$ is the predictor's function and $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$. For an observation instance $X = x_i$, we compute the mean square error loss of the prediction $\hat{f}(x_i)$:

$$\begin{aligned}
\text{MSE}(x_i) &= \text{E}\left[\left(\hat{f}(x_i) - y\right)^2\right]_{X=x_i} \\
&= \text{E}\left[\left(\hat{f}(x_i) - f(x_i) - \varepsilon\right)^2\right] && \text{Let } \mu = \text{E}\left(\hat{f}(x_i)\right) \\
&= \text{E}\left[\left(\hat{f}(x_i) - \mu + \mu - f(x_i) - \varepsilon\right)^2\right] \\
&= \underbrace{\text{E}\left(\hat{f}(x_i) - \mu\right)^2}_{\text{variance}} + \underbrace{\text{E}\Big(\mu - f(x_i)\Big)^2}_{\text{bias squared}} + \text{E}\Big(\varepsilon - 0\Big)^2 && \text{(rest coefficients are equal to 0)} \\
&= \text{var}\left(\hat{f}(x_i)\right) + \text{bias}\left(\hat{f}(x_i)\right)^2 + \sigma_\varepsilon^2.
\end{aligned} \tag{A.5}$$

In that way, the MSE is broken down into two main components: the variance of the prediction and the squared bias showing the amount by which the average of our estimate differs from the target value. The third component is the variance of the target around its true mean something that cannot be avoided unless σ_ε^2 is equal to 0. This process is known as bias-variance decomposition. Depending on their properties learning models face bias-variance tradeoffs. Commonly, increasing the complexity of a model, results to a lower bias but higher variance. See [67] for an extended reference on bias-variance decomposition.

A.3 Cosine Similarity

For two vectors x and y of the same size, their cosine similarity is simply the cosine of their angle θ_{xy}. Therefore

$$\text{Sim}(x, y) = \cos(\theta_{xy}) = \frac{x \cdot y}{\|x\|_{\ell_2} \|y\|_{\ell_2}}. \tag{A.6}$$

$\text{Sim}(x, y)$ ranges from -1 to 1 and intuitively the higher it is, the stronger the similarity between the vectors under comparison.

A.4 Moving Average

Given a series of k points $X = \{x_1, x_2, ..., x_k\}$, the n-point moving average is an equally sized series $X^{(n)} = \{x_1^*, x_2^*, ..., x_k^*\}$, where

$$x_j^* = \frac{x_{j-\lfloor n/2 \rfloor} + ... + x_{j-1} + x_j + x_{j+1} + ... + x_{j+\lfloor n/2 \rfloor}}{n}, \tag{A.7}$$

and n is an odd number. When $j \leq \lfloor n/2 \rfloor$ or $j > k - \lfloor n/2 \rfloor$, the enumerator of the above equation can have $2j - 1$ or $k - j + 1$ elements respectively.

In our work, we are applying the moving average method as a smoothing technique. A simple example follows. Suppose that we are adding random noise $\in$ [-1, 1] to the $\sin(x)$ function, where $x \in$ [-2π, 2π]. In figure A.1, we have plotted a 513-point sample of the $\sin(x)$ + **noise** signal, as well as its 64-point moving average. We see that by smoothing most of the (artificial) random noise has been removed recovering the original $\sin(x)$ signal.

A.5 Null hypothesis, p-value and statistical significance

A **null hypothesis** is an assertion, which is made on the expectation to be rejected due to the lack of statistical proof in the data that will support it. By rejecting a null hypothesis, we are aiming to support a hypothesis of an opposite nature, if such hypothesis can be defined.

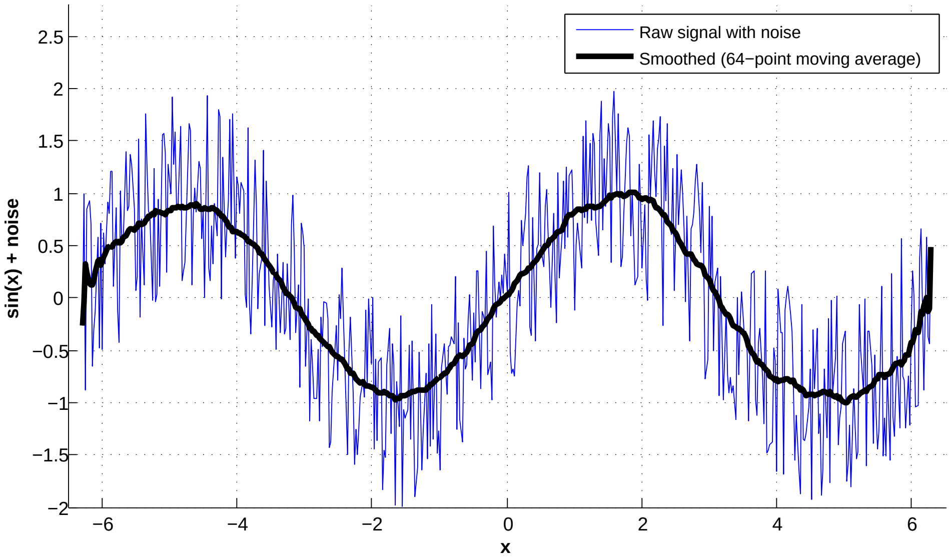


Figure A.1: Applying moving average to smooth and remove noise from a $\sin(x)$ signal.

P-value ρ is the probability that the null hypothesis is true. Given a set of n trials, ρ is equal to:

$$\rho = \frac{\#\{\text{Null hypothesis is true}\}}{n}, \tag{A.8}$$

i.e. the number of times that the null hypothesis is true divided by the number of trials.

If a p-value is below a threshold α – in most occasions $\alpha = 0.05$ –, then the null hypothesis can be rejected due to the fact that statistics indicate it is unlikely to hold. This usually serves as an indication that a hypothesis of an opposite nature or a related assumption, which contradicts with the null hypothesis, might be **statistically significant**.

A.6 Principal Component Analysis

A widely used technique with a clear application to feature extraction is Principal Component Analysis (**PCA**). Given X, a set of n D-dimensional observations $\{x_1, ..., x_n\}$, PCA can be defined as the orthogonal projection of X onto an M-dimensional space, where $M \leq D$, with respect to maximising the variance of the projected data [70].[1] Generally, it can be

[1] PCA was first introduced by Pearson in [125].

expressed as the following constrained maximisation problem:

$$\begin{aligned} \max_{w_1} \quad & w_1^{\mathrm{T}} S w_1 \\ \text{s.t.} \quad & w_1^{\mathrm{T}} w_1 \leq 1, \end{aligned} \tag{A.9}$$

where w_1 denotes the direction of the variance and S is the covariance matrix of the X, *i.e.*

$$S = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^{\mathrm{T}}. \tag{A.10}$$

It is shown that the solution of this optimisation problem is derived by solving a simple eigenvalue problem:

$$S w_1 = \lambda_1 w_1. \tag{A.11}$$

The eigenvector w_1 is known as the principal component of X; additional principal components can be defined by computing the M eigenvectors w_1, ..., w_M of S corresponding to the M largest eigenvalues λ_1, ..., λ_M [10].

Appendix B

Complementary Tables and Data

B.1 Urban Centres Per Region

Twitter data collection has been focused on a set of 54 UK urban centres. We collect tweets geolocated within a 10Km range from each centre. When inferring flu rates, we are using sets or groups of locations that belong in one geographic region. Here we list all the considered geographic regions as well as the locations belonging to each one of them.

- **South England**: Brighton, Bristol, Bournemouth, Gloucester, London, Oxford, Plymouth, Poole, Portsmouth, Reading, Slough, Southampton, Swindon
- **Central England**: Basildon, Birmingham, Coventry, Derby, Ipswich, Leicester, Luton, Northampton, Norwich, Nottingham, Peterborough, Southend, Stoke, Watford, Wolverhampton
- **North England**: Blackburn, Blackpool, Bradford, Bolton, Huddersfield, Hull, Leeds, Liverpool, Manchester, Middlesbrough, Newcastle, Oldham, Preston, Rotherham, Sheffield, Stockport, Sunderland, York
- **Scotland**: Aberdeen, Dundee, Edinburgh, Glasgow
- **Wales**: Cardiff, Newport, Swansea
- **Northern Ireland**: Belfast

B.2 An important observation – 41 flu markers

This is the list of terms (Table B.1) used in one of our very first experiments aiming to extract flu rates from Twitter content (Section 4.2.2). It is a manually formed list with 41 textual markers (or n-grams) that we thought they might be able to describe an illness (in that particular case a flu-like illness) and its symptoms.

Table B.1: 41 hand-picked flu markers

aches	fatigue	pain
back pain	feel sick	pneumonia
blood	feeling bad	runny nose
catarrh	feeling unwell	shivers
cold	fever	sick
colic	flu	sneezes
coma	gastroenteritis	sore throat
cough	headache	spasm
cramps	infection	stomach ache
croups	inflammation	temperature
diarrhea	irritation	tiredness
discharge	muscles ache	tonsils
dizzy	muscles pain	vomit
fainting	nausea	

B.3 Detecting temporal mood patterns – 1-grams from WordNet Affect

Tables with the terms used to track affective norms for anger (Table B.2), fear (Table B.3), joy (Table B.4) and sadness (Table B.5). Those terms have been retrieved from WordNet Affect [153] first by removing the very few existing 2-grams and then by stemming all 1-grams.

Table B.2: List of stemmed words used to detect **Anger** – 146 terms

abhor	abhorr	abomin	aggrav	aggress	amok	amuck	anger	angri	angrili	animos	animu
annoi	annoy	antagon	avarici	baffl	balk	bedevil	begrudg	belliger	bitter	bother	bothersom
brood	chafe	choler	covet	crucifi	dander	despis	despit	detest	devil	discourag	disdain
displeas	displeasur	dudgeon	dun	enmiti	enrag	envi	enviabl	enviou	envious	evil	exacerb
exasper	execr	fit	frustrat	furi	furiou	furious	gall	grabbi	grasp	gravel	greedi
grievanc	grizzl	grudg	hackl	harass	harri	hate	hatr	heartburn	hostil	huffi	huffili
incens	indign	indignantli	infuri	ir	irasc	irrit	jealou	jealous	jealousi	livid	loath
mad	madden	malefic	malevol	malic	malici	malign	misanthrop	misanthropi	misogyn	misogyni	mison
murder	nettl	nettlesom	odium	offend	offens	oppress	outrag	peev	persecut	peski	pester
pestifer	piqu	pout	prehensil	provok	rag	rage	rancor	resent	resentfulli	rile	roil
scene	score	scorn	sore	spite	spleen	stew	stung	sulk	sulki	tantal	tantrum
teas	temper	torment	umbrag	veng	vengefulli	venom	vex	vexat	vexati	vindict	warpath
wrath	wroth										

Table B.3: List of stemmed words used to detect **Fear** – 92 terms

affright	afraid	alarm	alert	anxiou	anxious	appal	apprehens	atroci	aw	awfulli	bashfulli
bode	browbeaten	bulli	chill	constern	cow	cower	crawl	creep	cring	cruel	cruelli
cruelti	dash	daunt	diffid	dire	dismai	dread	dreadfulli	fawn	fear	fearfulli	fearsom
forebod	fright	frighten	frighteningli	grovel	hangdog	hardheart	heartless	heartlessli	hesit	hesitantli	hesitatingli
hideou	hideous	horrend	horribl	horrid	horrif	horrifi	horrifyingli	horror	hyster	hysteria	intimid
merciless	monstrous	outrag	pall	panic	panick	panicki	pitiless	premonit	presag	presenti	ruthless
scare	scari	scarili	shadow	shiveri	shyli	shyness	suspens	terribl	terrifi	terror	timid
timidli	timor	trepid	ugli	unassert	uneasili	unkind	unsur				

Table B.4: List of stemmed words used to detect **Joy** – 224 terms

admir	ador	affect	affection	amatori	amic	amor	anticip	appreci	approb	approv
ardor	attach	avid	bang	banter	barrack	beam	beguil	belong	benef	benefic
benefici	benevol	bewitch	blith	bonheur	brotherhood	brotherli	buoyanc	captiv	captur	care
carefre	catch	chaff	charg	charit	charm	cheer	cheerfulli	cheeri	close	comfort
commend	compat	complac	concern	congratul	consol	content	crush	delight	devot	eager
eagerli	ebulli	elan	elat	embolden	emot	empath	empathet	empathi	enamor	enamour
enchant	endearingli	enjoi	enthral	enthusiasm	enthusiast	entranc	esteem	euphor	euphoria	exalt
excit	exhilar	exhort	expans	expect	exuber	exuberantli	exult	exultantli	fanci	fascin
favor	favour	festal	festiv	flush	fond	fondli	fratern	friendli	fulfil	gai
gaieti	gala	glad	gladden	glee	gleeful	gleefulli	gloat	good	goodwil	gratifi
gratifyingli	great	gusto	happi	happili	heart	hearten	hilar	hilari	identif	identifi
impress	infatu	insouci	inspir	intimaci	intox	jocular	jocund	joi	jolli	jolliti
jovial	joy	joyfulli	joyou	joyous	jubil	jubilantli	keen	kick	kid	kindheart
kindli	laudabl	lighthearted	lightsom	likabl	like	likeabl	love	loving	lovingli	loyalti
merri	merrili	merriment	mirth	mirthfulli	near	nigh	occupi	partial	penchant	perki
predilect	preen	prefer	pride	protect	proudli	rapport	recreat	regard	rejoic	relish
respect	revel	riotous	romant	rush	satisfact	satisfactori	satisfactorili	satisfi	satisfyingli	schadenfreud
scream	smug	softheart	solac	sooth	stimul	strike	sunni	suspens	sympathet	sympathi
tast	teas	tender	thrill	tickl	titil	tranc	triumph	triumphal	triumphant	triumphantli
unworri	uplift	uproari	urg	wallow	warm	warmheart	warmhearted	warmth	weak	worri
worship	zeal	zealou	zest							

B.4 Inferring Voting Intentions from Twitter Content – Search Terms

Terms used to select tweets related to the major political parties in the UK. Character '_' denotes an empty space and '#' denotes a Twitter topic. Tables B.6, B.7 and B.8 hold the terms for the Conservative (52 terms), Labour (57 terms) and Liberal Democrat party (62 terms) respectively.

Table B.5: List of stemmed words used to detect **Sadness** – 115 terms

aggriev	attrit	bad	bereav	bereft	blue	bore	cheerless	compunct	contrit	dark
deject	demor	deplor	depress	desol	despairingli	despond	dingi	disconsol	discourag	dishearten
dismai	dismal	dispirit	distress	dole	dolefulli	dolor	downcast	downheart	downtrodden	drab
drear	dreari	dysphor	dysphoria	execr	forlorn	forlornli	gloom	gloomi	gloomili	glum
godforsaken	grief	griev	grievou	grievous	grim	guilt	guilti	hangdog	hapless	harass
heartach	heartbreak	heartrend	heartsick	helpless	joyless	joylessli	lachrymos	laden	lament	loneli
lorn	low	melanchol	melancholi	miser	miseri	misfortun	mourn	mournfulli	oppress	pathet
penanc	penit	penitenti	persecut	piteou	piti	pitiabl	plaintiv	poor	regret	remors
remorsefulli	repent	rue	rueful	ruefulli	ruth	sad	sadden	sadli	shame	shamefac
somber	sorri	sorrow	sorrowfulli	suffer	tear	tyrann	unhappi	weep	weepi	weight
woe	woebegon	woeful	woefulli	wretch						

Table B.6: Terms for the Conservative Party (**CON**) – 52 terms

#conservative	CONSERVATIVE	Eric Pickles	Margaret Thatcher
#Tories	Conservative	Francis Maude	Michael Gove
#tories	Conservative Party	George Osborne	Oliver Letwin
#Tory	Crispin Blunt	Gerald Howarth	Owen Paterson
#tory	David Cameron	Grant Shapps	Patrick McLoughlin
_TORIES	David Davis	Hugo Swire	Philip Hammond
_tories	David Mundell	Iain Smith	President Cameron
Alan Duncan	David Willetts	Jeremy Hunt	Theresa May
Andrew Lansley	debate Cameron	John Redwood	Tories
Andrew Mitchell	Dominic Grieve	Ken Clarke	vote Cameron
Boris Johnson	Duncan Smith	Lady Warsi	vote conservative
Caroline Spelman	Edward Leigh	Leader Cameron	vote Tory
Cheryl Gillan	Edward Vaizey	Liam Fox	William Hague

Table B.7: Terms for the Labour Party (**LAB**) – 57 terms

#labour	debate Brown	LABOUR	Rosie Winterton
_Ed _Balls	Diane Abbott	Labour	Sadiq Khan
Alan Johnson	Douglas Alexander	Labour Party	Shaun Woodward
Alastair Campbell	Ed Miliband	Leader Brown	Stella Creasy
Andy Burnham	Gordon Brown	Liam Byrne	Tessa Jowell
Angela Eagle	Government Brown	Luciana Berger	Tom Harris
Ann McKechin	Harriet Harman	Maria Eagle	Tom Watson
Ben Bradshaw	Hilary Benn	Mary Creagh	Tony Blair
Caroline Flint	Ivan Lewis	Meg Hillier	Tony Lloyd
Charles Clarke	Jim Murphy	MILIBAND	vote Brown
Chris Bryant	John Denham	Miliband	vote labour
Chuka Umunna	John Healey	Peter Hain	Yvette Cooper
David Blunkett	John Prescott	Peter Mandelson	
David Lammy	Jon Trickett	President Brown	
David Miliband	Kerry McCarthy	Prime Minister Brown	

Table B.8: Terms for the Liberal Democrat Party (**LIB**) – 62 terms

#Clegg
#clegg
#liberal
_Ed Davey
_Jo Swinson
Alistair Carmichael
Andrew Stunell
Charles Kennedy
Chris Huhne
Dan Rogerson
Danny Alexander
David Heath
David Shutt
debate Clegg
Duncan Hames
Edward Davey
Fiona Hall
Greg Mulholland
Jenny Willott
Jeremy Browne
John Hemming
John Pugh
John Thurso
Julian Huppert
Leader Clegg
Lembit Opik
Lib_ _Dem
LIBDEM
LibDem
Libdem
libDem
libdem
LIBERAL
Liberal
Liberal Democrat
Lord McNally
Lord Shutt
Lorely Burt
Lynne Featherstone
MP Clegg
Mark Hunter
Martin Horwood
Michael Crockart
Mike Crockart
Mike Hancock
Nicholas Clegg
Nick Clegg
Nick Harvey
Norman Backer
Norman Lamb
Paul Burstow
President Clegg
Sarah Teather
Simon Hughes
Steve Webb
Tessa Munt
Tim Farron
Tom McNally
Vince Cable
Vincent Cable
vote Clegg
vote liberal

Bibliography

[1] C. Apte, F. Damerau, and S. Weiss. *Text Mining with Decision Rules and Decision Trees*. Tech. rep. June. IBM Research, 1998, p. 4.

[2] S. Asur and B. A. Huberman. “Predicting the Future with Social Media”. In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE, Aug. 2010, pp. 492–499.

[3] S. Baccianella, A. Esuli, and F. Sebastiani. “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining”. In: *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC)*. ELRA, 2010, pp. 2200–2204.

[4] F. R. Bach. “Bolasso: Model Consistent Lasso Estimation through the Bootstrap”. In: *Proceedings of the 25th International Conference on Machine learning (ICML)* (2008), pp. 33–40.

[5] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999, p. 513.

[6] M. Banbura, D. Giannone, and L. Reichlin. *Nowcasting*. Tech. rep. Universite Libre de Bruxelles, 2010.

[7] P. L. Bartlett, S. Mendelson, and J. Neeman. “L1-regularized linear regression: persistence and oracle inequalities”. In: *Probability Theory and Related Fields* (June 2011), pp. 1–32.

[8] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995, p. 791.

[9] P. J. Bickel, Y. Ritov, and A. B. Tsybakov. “Simultaneous analysis of Lasso and Dantzig selector”. In: *Annals of Statistics* 37.4 (2008), pp. 1705–1732.

[10] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] J. Bollen, H. Mao, and X. Zeng. “Twitter mood predicts the stock market”. In: *Journal of Computational Science* 2.1 (Feb. 2011), pp. 1–8.

[12] K. Borau et al. “Microblogging for Language Learning: Using Twitter to Train Communicative and Cultural Competence”. In: *Applied Linguistics* 5686.500 (2009), pp. 78–87.

[13] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling*. Springer, 2005, p. 614.

[14] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004, p. 730.

[15] A. Branch and K. Fukunaga. “A Branch and Bound Algorithm for Feature Subset Selection”. In: *IEEE Transactions on Computers* C.9 (1977), pp. 917–922.

[16] L. Breiman et al. *Classification and Regression Trees*. Wadsworth, 1984, p. 368.

[17] L. Breiman. “Bagging predictors”. In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140.

[18] L. Breiman. “Heuristics of instability and stabilization in model selection”. In: *Annals of Statistics* 24.6 (1996), pp. 2350–2383.

[19] P. Bühlmann and B. Yu. “Analyzing bagging”. In: *The Annals of Statistics* 30.4 (Aug. 2002), pp. 927–961.

[20] A. Buja and W. Stuetzle. “Smoothing Effects of Bagging”. In: *AT&T Labs Research* (2000).

[21] A. Buja and W. Stuetzle. “The Effect of Bagging on Variance, Bias, and Mean Squared Error”. In: *AT&T Labs Research* (2000).

[22] R. A. Calvo and S. D’Mello. “Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications”. In: *IEEE Transactions on Affective Computing* 1.1 (Jan. 2010), pp. 18–37.

[23] E. J. Candès and Y. Plan. “Near-ideal model selection by l1 minimization”. In: *Annals of Statistics* 37.5A (2009), pp. 2145–2177.

[24] M. Chen et al. “Failure diagnosis using decision trees”. In: *International Conference on Autonomic Computing* (2004), pp. 36–43.

[25] W. S. Cleveland. “Robust Locally Weighted Regression and Smoothing Scatterplots”. In: *Journal of the American Statistical Association* 74.368 (1979), pp. 829–836.

[26] N. Collier, N. Son, and N. Nguyen. “OMG U got flu? Analysis of shared health messages for bio-surveillance”. In: *Journal of Biomedical Semantics* 2.Suppl 5:S9 (Oct. 2011).

[27] B. O. Connor et al. “From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series”. In: *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*. May. AAAI Press, 2010, pp. 122–129.

[28] M. D. Conover et al. “Political Polarization on Twitter”. In: *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*. AAAI Press, 2011, pp. 89–96.

[29] C. D. Corley et al. “Monitoring Influenza Trends through Mining Social Media”. In: *International Conference on Bioinformatics & Computational Biology* (2009), pp. 340–346.

[30] A. Culotta. *Detecting influenza outbreaks by analyzing Twitter messages*. Tech. rep. Southeastern Louisiana University, 2010, p. 11.

[31] A. Culotta. “Towards detecting influenza epidemics by analyzing Twitter messages”. In: *Proceedings of the 1st Workshop on Social Media Analytics (SOMA)*. ACM, 2010, pp. 115–122.

[32] B. Debatin et al. “Facebook and Online Privacy: Attitudes, Behaviors, and Unintended Consequences”. In: *Journal of Computer-Mediated Communication* 15.1 (2009), pp. 83–108.

[33] N. A. Diakopoulos and D. A. Shamma. “Characterizing debate performance via aggregated twitter sentiment”. In: *Proceedings of the 28th International Conference on Human factors in Computing Systems (CHI)*. ACM Press, 2010, pp. 1195–1198.

[34] S. Doan, B.-K. H. Vo, and N. Collier. “An analysis of Twitter messages in the 2011 Tohoku Earthquake”. In: *CoRR*. arXiv:1109.1618 (2011), p. 9.

[35] N. R. Draper and H. Smith. *Applied Regression Analysis*. John Wiley & Sons, 1998, p. 706.

[36] J. Dunlap and P. Lowenthal. “Tweeting the Night Away: Using Twitter to Enhance Social Presence”. In: *Journal of Information Systems Education* 20.2 (2009), pp. 129–136.

[37] B. Efron. “Bootstrap Methods: Another Look at the Jackknife”. In: *The Annals of Statistics* 7.1 (1979), pp. 1–26.

[38] B. Efron and R. Tibshirani. “Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy”. In: *Statistical Science* 1.1 (1986), pp. 54–75.

[39] B. Efron et al. “Least angle regression”. In: *The Annals of statistics* 32.2 (2004), pp. 407–451.

[40] B. Efron and R. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall, 1993, p. 436.

[41] A. J. Elliot et al. “Monitoring the emergence of community transmission of influenza A/H1N1 2009 in England: a cross sectional opportunistic survey of self sampled telephone callers to NHS Direct”. In: *British Medical Journal* 339 (2009), b3403.

[42] A. Esuli and F. Sebastiani. “SentiWordNet: A publicly available lexical resource for opinion mining”. In: *Proceeding of the 5th Conference on Language Resources and Evaluation (LREC)*. 2006, pp. 417–422.

[43] G. Eysenbach. “Infodemiology and infoveillance: framework for an emerging set of public health informatics methods to analyze search, communication and publication behavior on the Internet.” In: *Journal of medical Internet research*. 11(1):e11 (Jan. 2009).

[44] Facebook Inc. *S-1 Registration Statement*. Tech. rep. (http://goo.gl/Y2LNw), Securities and Exchange Commission, 2012.

[45] I. N. Flaounas, N. Fyson, and N. Cristianini. “Predicting Relations in News-Media Content among EU Countries”. In: *Proceedings of 2nd Workshop on Cognitive Information Processing (CIP)* (2010), pp. 269–274.

[46] D. M. Fleming and A. J. Elliot. “Lessons from 40 years’ surveillance of influenza in England and Wales”. In: *Epidemiology and Infection* 136.7 (2007), pp. 866–875.

[47] G. C. Fonarow et al. “Risk stratification for in-hospital mortality in acutely decompensated heart failure: classification and regression tree analysis.” In: *Journal Of The American Medical Association (JAMA)* 293.5 (2005), pp. 572–580.

[48] C. Fox. “A stop list for general text”. In: *SIGIR Forum* 24.1-2 (Sept. 1989), pp. 19–21.

[49] W. B. Frakes. “Stemming algorithms”. In: *Information Retrieval*. Prentice-Hall, 1992. Chap. 8, pp. 131–160.

[50] J. Friedman and P. Hall. “On bagging and nonlinear estimation”. In: *Journal of Statistical Planning and Inference* 137.3 (Mar. 2007), pp. 669–683.

[51] G. M. Furnival and R. W. Wilson. “Regression by leaps and bounds”. In: *Technometrics* 16.4 (1974), pp. 499–511.

[52] J. Furnkranz. *A study using n-gram features for text categorization*. Tech. rep. Austrian Research Institute for Artificial Intelligence, 1998, p. 10.

[53] D. Garlaschelli and M. I. Loffredo. “Patterns of link reciprocity in directed networks”. In: *Physical Review Letters* 93.26 (2004), pp. 1–4.

[54] D. Gayo-Avello, P. T. Metaxas, and E. Mustafaraj. “Limits of Electoral Predictions using Twitter”. In: *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*. 2011, pp. 490–493.

[55] D. Giannone, L. Reichlin, and D. Small. “Nowcasting: The real-time informational content of macroeconomic data”. In: *Journal of Monetary Economics* 55.4 (2008), pp. 665–676.

[56] J. Ginsberg et al. “Detecting influenza epidemics using search engine query data”. In: *Nature* 457.7232 (2008), pp. 1012–1014.

[57] S. A. Golder and M. W. Macy. “Diurnal and Seasonal Mood Vary with Work, Sleep, and Daylength Across Diverse Cultures”. In: *Science* 333.6051 (2011), pp. 1878–1881.

[58] J. Gomide et al. “Dengue surveillance based on a computational model of spatio-temporal locality of Twitter”. In: *Proceedings of the ACM WebSci*. ACM, 2011, pp. 1–8.

[59] E. Greenshtein. “Best subset selection, persistence in high-dimensional statistical learning and optimization under l1 constraint”. In: *Annals of Statistics* 34.5 (2007), pp. 2367–2386.

[60] S. Grimaldi et al. “Seasonal vegetative and affective symptoms in the Finnish general population: Testing the dual vulnerability and latitude effect hypotheses.” In: *Nordic Journal of Psychiatry* 63.5 (Apr. 2009), pp. 397–404.

[61] G. Grosseck and C. Holotescu. “Can we use Twitter for educational activities?” In: *4th International Scientific Conference e-Learning and Software for Education (eLSE)*. 2008, p. 11.

[62] A. Gulli and A. Signorini. “The indexable web is more than 11.5 billion pages”. In: *Special interest tracks and posters of the 14th international conference on World Wide Web (WWW)*. ACM, 2005, pp. 902–903.

[63] I. Guyon and A. Elisseeff. “An Introduction to Variable and Feature Selection”. In: *Journal of Machine Learning Research* 3.7-8 (2003), pp. 1157–1182.

[64] I. Guyon et al., eds. *Feature Extraction*. Vol. 207. Studies in Fuzziness and Soft Computing. Springer, 2006.

[65] J. Han, M. Kamber, and J. Pei. *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2011, p. 550.

[66] B. P. Hasler et al. “Preliminary evidence of diurnal rhythms in everyday behaviors associated with positive affect”. In: *Journal of Research in Personality* 42.6 (Dec. 2008), pp. 1537–1546.

[67] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2005, p. 552.

[68] A. E. Hoerl and R. W. Kennard. “Ridge regression: Biased estimation for nonorthogonal problems”. In: *Technometrics* 12.1 (1970), pp. 55–67.

[69] C. Honeycutt and S. C. Herring. “Beyond Microblogging: Conversation and Collaboration via Twitter”. In: *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS)*. IEEE Press, 2009, pp. 1–10.

[70] H. Hotelling. “Analysis of a complex of statistical variables into principal components”. In: *Journal of Educational Psychology* 24.6 (1933), pp. 417–441.

[71] B. A. Huberman, D. M. Romero, and F. Wu. “Social networks that matter: Twitter under the microscope”. In: *First Monday* 14.1 (2008), pp. 1–9.

[72] A. L. Hughes and L. Palen. “Twitter adoption and use in mass convergence and emergency events”. In: *International Journal Of Emergency Management* 6.3-4 (2009), pp. 248–260.

[73] L. Hyafil and R. L. Rivest. “Constructing optimal binary decision trees is NP-complete”. In: *Information Processing Letters* 5.1 (1976), pp. 15–17.

[74] D. Ivnitski et al. “Biosensors for detection of pathogenic bacteria”. In: *Biosensors and Bioelectronics* 14.7 (1999), pp. 599–624.

[75] A. K. Jain, R. P. W. Duin, and J. Mao. “Statistical pattern recognition: A review”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.1 (2000), pp. 4–37.

[76] B. J. Jansen et al. “Twitter power: Tweets as electronic word of mouth”. In: *Journal of the American Society for Information Science and Technology* 60.11 (2009), pp. 2169–2188.

[77] A. Java et al. “Why we twitter”. In: *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis - WebKDD/SNA-KDD ’07*. Computer Science and Electrical Engineering. ACM, 2007, pp. 56–65.

[78] G. J. Jenkins, M. C. Perry, and M. J. Prior. *The climate of the United Kingdom and recent trends*. Met Office Hadley Centre Exeter, UK, 2008.

[79] H. Jick et al. “The United Kingdom 2009 Swine Flu Outbreak As Recorded in Real Time by General Practitioners”. In: *Epidemiology Research International* 2011.381597 (2011), pp. 1–6.

[80] G. V. Kass. “An Exploratory Technique for Investigating Large Quantities of Categorical Data”. In: *Applied Statistics* 29.2 (1980), pp. 119–127.

[81] P. Kokol et al. “Decision trees based on automatic learning and their use in cardiology”. In: *Journal of Medical Systems* 18.4 (1994), pp. 201–206.

[82] V. Koltchinskii. “Sparsity in penalized empirical risk minimization”. In: *Annales de lInstitut Henri Poincaré Probabilités et Statistiques* 45.1 (2009), pp. 7–57.

[83] G. Kowalski. *Information Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers, 1997, p. 296.

[84] M. Krieck et al. “A New Age of Public Health: Identifying Disease Outbreaks by Analyzing Tweets”. In: *3rd International Conference on Web Science*. 2011.

[85] B. Krishnamurthy, P. Gill, and M. Arlitt. “A few chirps about twitter”. In: *Proceedings of the first workshop on Online social networks - WOSP ’08* (2008), p. 19.

[86] N. Kronfeld-Schor and H. Einat. “Circadian rhythms and depression: Human psychopathology and animal models.” In: *Neuropharmacology* 62.1 (2012), pp. 101–14.

[87] H. Kwak et al. “What is Twitter, a social network or a news media?” In: *Proceedings of the 19th international conference on World Wide Web (WWW)*. ACM, 2010, pp. 591–600.

[88] V. Lampos. *On voting intentions inference from Twitter content: a case study on UK 2010 General Election*. Tech. rep. University of Bristol, Apr. 2012, p. 11.

[89] V. Lampos. *Weather Talk - extracting weather information by text mining*. Tech. rep. University of Bristol, 2008, p. 67.

[90] V. Lampos and N. Cristianini. “Diagnosing flu symptoms with social media”. In: *Natural Hazards Observer* XXXVI.4 (Mar. 2012), pp. 7–9.

[91] V. Lampos and N. Cristianini. “Nowcasting Events from the Social Web with Statistical Learning”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 3.4 (Sept. 2011).

[92] V. Lampos and N. Cristianini. “Tracking the flu pandemic by monitoring the social web”. In: *Proceedings of the 2nd International Workshop on Cognitive Information Processing (CIP)*. IEEE Press, June 2010, pp. 411–416.

[93] V. Lampos, T. De Bie, and N. Cristianini. “Flu Detector - Tracking Epidemics on Twitter”. In: *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*. Springer, 2010, pp. 599–602.

[94] V. Lampos et al. “Detecting Temporal Mood Patterns in the UK by Analysis of Twitter Content”. In: *British Journal of Psychiatry* (2012).

[95] T. Lansdall-Welfare, V. Lampos, and N. Cristianini. “Effects of the Recession on Public Mood in the UK”. In: *Mining Social Network Dynamics (MSND) session on Social Media Applications in News and Entertainment (SMANE) at WWW ’12*. Lyon, France: ACM, 2012, pp. 1221–1226.

[96] T. S. Lee et al. “Mining the customer credit using classification and regression tree and multivariate adaptive regression splines”. In: *Computational statistics and data analysis* 50.4 (2006), pp. 1113–1130.

[97] R. J. Lewis. *An Introduction to Classification and Regression Tree (CART) Analysis*. Tech. rep. 310. Harbor-UCLA Medical Center, 2000, p. 14.

[98] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer, 2007, p. 600.

[99] W.-Y. Loh and Y.-S. Shih. “Split Selection Methods for Classification Trees”. In: *Statistica Sinica* 7 (1997), pp. 815–840.

[100] K. Lounici. “Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators”. In: *Electronic Journal of Statistics* 2 (2008), pp. 90–102.

[101] J. B. Lovins. “Development of a stemming algorithm”. In: *Mechanical Translation and Computational Linguistics* 11.1-2 (1968), pp. 22–31.

[102] C. Lui, P. T. Metaxas, and E. Mustafaraj. “On the predictability of the U.S. elections through search volume activity”. In: *Proceedings of the IADIS International Conference on e-Society*. 2011.

[103] J. Lv and Y. Fan. “A unified approach to model selection and sparse recovery using regularized least squares”. In: *The Annals of Statistics* 37.6A (Dec. 2009), pp. 3498–3528.

[104] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008, p. 544.

[105] W. L. Martinez and A. R. Martinez. *Computational Statistics Handbook with Matlab*. Vol. 99. Chapman & Hall/CRC Computer Science & Data Analysis 466. Chapman & Hall, 2002, p. 616.

[106] N. Meinshausen. “Relaxed Lasso”. In: *Computational Statistics & Data Analysis* 52.1 (2007), pp. 374–393.

[107] N. Meinshausen and P. Bühlmann. “Stability selection”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.4 (July 2010), pp. 417–473.

[108] W. P. Menzel and C. H. Frances. “Application of GOES-8/9 Soundings to Weather Forecasting and Nowcasting”. In: *Bulletin of the American Meteorological Society* 79 (1998), pp. 2059–2077.

[109] P. T. Metaxas, E. Mustafaraj, and D. Gayo-Avello. “How (Not) To Predict Elections”. In: *IEEE 3rd International Conference on Social Computing (SocialCom)*. IEEE Press, 2011, pp. 165 –171.

[110] T. G. Metcalf, J. L. Melnick, and M. K. Estes. “Environmental virology: from detection of virus in sewage and water by isolation to identification by molecular biology-a trip of over 50 years”. In: *Annual Reviews in Microbiology* 49.1 (1995), pp. 461–487.

[111] G. A. Miller. “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[112] R. Milo et al. “Network motifs: simple building blocks of complex networks.” In: *Science* 298.5594 (2002), pp. 824–827.

[113] A. Mislove et al. “Understanding the Demographics of Twitter Users”. In: *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*. AAAI, 2011, pp. 554–557.

[114] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997, p. 432.

[115] T. M. Mitchell. *The Discipline of Machine Learning*. Tech. rep. Carnegie Mellon University, 2006, p. 7.

[116] P Mooney and P Corcoran. “Integrating volunteered geographic information into pervasive health computing applications”. In: *5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. IEEE Press, 2011, pp. 93–100.

[117] G. Murray, N. Allen, and J. Trinder. “Mood and the circadian system: investigation of a circadian component in Positive Affect”. In: *Chronobiology International* 19.6 (2002), pp. 1151–1169.

[118] S. K. Murthy. “Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey”. In: *Data Mining and Knowledge Discovery* 2.4 (1998), pp. 345–389.

[119] K. M. Neuzil, C. Hohlbein, and Y. Zhu. “Illness among schoolchildren during influenza season: effect on school absenteeism, parental absenteeism from work, and secondary illness in families”. In: *Archives of pediatrics & adolescent medicine* 156.10 (2002), pp. 986–991.

[120] B. O’Connor et al. “From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series”. In: *Proceedings of the International AAAI Conference on Weblogs and Social Media*. May. AAAI Press, 2010, pp. 122–129.

[121] A. Pak and P. Paroubek. “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”. In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)* (2010), pp. 1320–1326.

[122] B. Pang and L. Lee. “Opinion Mining and Sentiment Analysis”. In: *Foundations and Trends in Information Retrieval* 2.1-2 (2008), pp. 1–135.

[123] M. J. Paul and M. Dredze. *A Model for Mining Public Health Topics from Twitter*. Tech. rep. May 2009. Johns Hopkins University, 2010, p. 7.

[124] M. J. Paul and M. Dredze. “You Are What You Tweet: Analyzing Twitter for Public Health”. In: *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media* (2011), pp. 265–272.

[125] K. Pearson. “On lines and planes of closest fit to systems of points in space”. In: *Philosophical Magazine Series 6* 2.11 (1901), pp. 559–572.

[126] J. W. Pennebaker et al. *The Development and Psychometric Properties of LIWC2007*. Tech. rep. The University of Texas at Austin & University of Auckland, New Zealand, 2007, p. 22.

[127] P. M. Polgreen et al. “Using internet searches for influenza surveillance”. In: *Clinical Infectious Diseases* 47.11 (2008), pp. 1443–1448.

[128] M. A. Poole and P. N. O’Farrell. “The Assumptions of Linear Regression”. In: *Transactions of the Institute of British Geographers* 52 (Mar. 1971), pp. 145–158.

[129] M. F. Porter. “An algorithm for suffix stripping”. In: *Program* 14.3 (1980), pp. 130–137.

[130] M. F. Porter. *Snowball: A language for stemming algorithms*. Tech. rep. http://goo.gl/VYLB7, 2001.

[131] A. M. Prasad, L. R. Iverson, and A. Liaw. “Newer Classification and Regression Tree Techniques: Bagging and Random Forests for Ecological Prediction”. In: *Ecosystems* 9.2 (2006), pp. 181–199.

[132] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993, p. 302.

[133] J. R. Quinlan. “Induction of decision trees”. In: *Machine Learning* 1.1 (1986), pp. 81–106.

[134] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[135] W. Reinhardt et al. “How People are using Twitter during Conferences”. In: *Proceedings of 5th EduMedia Conference*. March. 2009, pp. 145–156.

[136] M. D. Riley. “Some applications of tree-based modelling to speech and language”. In: *Proceedings of the workshop on Speech and Natural Language (HLT)*. 2. Association for Computational Linguistics, 1989, pp. 339–352.

[137] L. Rokach and O. Z. Maimon. *Data mining with decision trees: theory and applications*. World Scientific Publishing, 2008, p. 244.

[138] C. L. Rusting and R. J. Larsen. “Diurnal Patterns of Unpleasant Mood: Associations with Neuroticism, Depression, and Anxiety”. In: *Journal of Personality* 66.1 (1998), pp. 85–103.

[139] T. Sakaki, M. Okazaki, and Y. Matsuo. "Earthquake shakes Twitter users: real-time event detection by social sensors". In: *Proceedings of the 19th international conference on World Wide Web (WWW)*. ACM, 2010, pp. 851–860.

[140] G. Salton and C. S. Yang. "On the specification of term values in automatic indexing". In: *Journal of documentation* 29.4 (1973), pp. 351–372.

[141] G. Salton, E. A. Fox, and H. Wu. "Extended Boolean information retrieval". In: *Communications of the ACM* 26.11 (Nov. 1983), pp. 1022–1036.

[142] A. L. Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229.

[143] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers II - Recent Progress". In: *IBM Journal of Research and Development* 11.6 (1967), pp. 601–617.

[144] J. Sankaranarayanan et al. "TwitterStand: news in tweets". In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*. ACM, 2009. Chap. GIS '09, pp. 42–51.

[145] N. Savage. "Twitter as medium and message". In: *Communications of the ACM* 54.3 (Mar. 2011), pp. 18–20.

[146] H. Schmid. "Probabilistic Part-of-Speech Tagging Using Decision Trees". In: *Proceedings of International Conference on New Methods in Language Processing*. Vol. 12. University of Stuttgart. 1994, pp. 44–49.

[147] C. E. Shannon. "Prediction and Entropy of Printed English". In: *Bell System Technical Journal* 30.1 (1951), pp. 50–64.

[148] A. Signorini, A. M. Segre, and P. M. Polgreen. "The use of Twitter to track levels of disease activity and public concern in the U.S. during the influenza A H1N1 pandemic." In: *PLoS ONE* 6.5 (2011), e19467.

[149] C. Silva and B. Ribeiro. "The importance of stop word removal on recall values in text categorization". In: *Proceedings of the International Joint Conference on Neural Networks*. Vol. 3. IEEE Press, 2003, pp. 1661–1666.

[150] C. J. Sims et al. "Predicting cesarean delivery with decision tree models". In: *American Journal of Obstetrics and Gynecology* 183.5 (2000), pp. 1198–1206.

[151] H. Smid et al. “Canary in a coal mine: monitoring air quality and detecting environmental incidents by harvesting Twitter”. In: *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems (CHI EA)*. ACM, 2011, pp. 1855–1860.

[152] A. A. Stone et al. “A population approach to the study of emotion: diurnal rhythms of a working day examined with the Day Reconstruction Method.” In: *Emotion (Washington, D.C.)* 6.1 (Feb. 2006), pp. 139–149.

[153] C. Strapparava and A. Valitutti. “WordNet-Affect: an affective extension of WordNet”. In: *Proceedings of LREC*. Vol. 4. ELRA. 2004, pp. 1083–1086.

[154] C. Strapparava and R. Mihalcea. “Learning to identify emotions in text”. In: *Proceedings of the 2008 ACM symposium on Applied computing (SAC)*. ACM, 2008, pp. 1556–1560.

[155] C Sutton. “Classification and Regression Trees, Bagging, and Boosting”. In: *Handbook of Statistics* 24.04 (2005), pp. 303–329.

[156] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Vol. 9. 5. MIT Press, 1998. Chap. Evaluative, pp. 1054–1054.

[157] C. Tan. “The use of bigrams to enhance text categorization”. In: *Information Processing & Management* 38.4 (2002), pp. 529–546.

[158] R. Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288.

[159] R Tibshirani et al. “Sparsity and smoothness via the fused lasso”. In: *Journal of the Royal Statistical Society Series B (Statistical Methodology)* 67.1 (2005), pp. 91–108.

[160] R. J. Tibshirani. “The Lasso Problem and Uniqueness”. In: (June 2012), p. 25.

[161] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. 2. W. H. Winston, 1977, p. 259.

[162] K. Toutanova and C. D. Manning. “Enriching the knowledge sources used in a maximum entropy part-of-speech tagger”. In: *Proceedings of the Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics* 13 (2000), pp. 63–70.

[163] K. Toutanova et al. *Feature-rich part-of-speech tagging with a cyclic dependency network*. Vol. 1. Association for Computational Linguistics, 2003, pp. 173–180.

[164] A. Tumasjan et al. “Predicting elections with Twitter: What 140 characters reveal about political sentiment”. In: *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*. AAAI, 2010, pp. 178–185.

[165] E. Tuv et al. “Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination”. In: *The Journal of Machine Learning Research* 10 (Dec. 2009), pp. 1341–1366.

[166] S. Vieweg et al. “Microblogging during two natural hazards events”. In: *Proceedings of the 28th international conference on Human factors in computing systems (CHI)*. ACM. ACM, 2010, pp. 1079–1088.

[167] N. Vora. “The viral storm: the dawn of a new pandemic age”. In: *The Lancet Infectious Diseases* 12.3 (Mar. 2012), p. 190.

[168] M. J. Wainwright. “Sharp thresholds for high-dimensional and noisy sparsity recovery using l1 -constrained quadratic programmming (Lasso)”. In: *IEEE Transactions on Information Theory* 55.5 (2009), pp. 2183–2202.

[169] L. Wasserman and K. Roeder. “High-dimensional variable selection”. In: *Annals of Statistics* 37.5A (2007), pp. 2178–2201.

[170] J. W. Wilson et al. “Nowcasting Thunderstorms: A Status Report”. In: *Bulletin of the American Meteorological Society* 79.10 (1998), pp. 2079–2099.

[171] M. Yuan and Y. Lin. “Model selection and estimation in regression with grouped variables”. In: *Journal of the Royal Statistical Society - Series B: Statistical Methodology* 68.1 (2006), pp. 49–67.

[172] D. Zhao and M. B. Rosson. “How and why people Twitter”. In: *Proceedings of the ACM 2009 international conference on Supporting group work (GROUP)*. ACM, 2009, pp. 243–252.

[173] P. Zhao and B. Yu. “On model selection consistency of Lasso”. In: *Journal of Machine Learning Research* 7.11 (2006), pp. 2541–2563.

[174] H. Zou. “The Adaptive Lasso and Its Oracle Properties”. In: *Journal of the American Statistical Association* 101.476 (2006), pp. 1418–1429.

Index