Nowcasting Events from the Social Web with Statistical Learning

VASILEIOS LAMPOS and NELLO CRISTIANINI, University of Bristol, UK

We present 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. Having geotagged 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 Influenza-like Illness 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, especially given the short length –approximately one year– of Twitter's data time series.

Categories and Subject Descriptors: G.3 [**Probability and Statistics**]: *Statistical computing*; I.2.6 [**Artificial Intelligence**]: Learning; I.5.4 [**Pattern Recognition**]: Applications—*Text processing*; J.3 [**Life and Medical Sciences**]: *Medical information systems*

General Terms: Algorithms, Design, Experimentation, Measurement, Performance

Additional Key Words and Phrases: Event detection, feature selection, LASSO, social network mining, sparse learning, Twitter

ACM Reference Format:
Lampos, V. and Cristianini, N. 2012. Nowcasting events from the social web with statistical learning. ACM Trans. Intell. Syst. Technol. 3, 4, Article 72 (September 2012), 22 pages.
DOI = 10.1145/2337542.2337557 http://doi.acm.org/10.1145/2337542.2337557

1. INTRODUCTION

It has not been a long time since snapshots of real life started to appear on the social side of the Web. Social networks such as Facebook and Twitter have grown stronger, forming an electronic substitute for public expression and interaction. Twitter, in particular, counting a total of 200 million users worldwide,[1] came up with a convention that encouraged users to make their posts, commonly known as *tweets*, by default publicly available. Tweets being limited to a length of 140 characters (similarly to text messages in mobile phones) forced their authors to produce more topic specific statements.

[1]Based on an email update titled as 'Get the most out of Twitter in 2011', sent by Twitter Inc. to its users (February 1, 2011).

V. Lampos would like to thank NOKIA Research, EPSRC (DTA/SB1826) and the Computer Science Department (University of Bristol) for all the various levels of support. N. Cristianini is supported by a Royal Society Wolfson Merit Award.
Authors' address: V. Lampos and N. Cristianini, Level 0, Intelligent Systems Laboratory, Merchant Venturers Building, Woodland Road, BS8 1UB, Bristol, UK; email: bill.lampos@gmail.com.

DOI 10.1145/2337542.2337557 http://doi.acm.org/10.1145/2337542.2337557

By adding user's location when posting (via the mobile phone service provider or IP address) to this piece of public information, Twitter ushered in a new era for social Web media and at the same time enabled a new wave of experimentation and research on text stream mining. Now, it has been shown that this vast amount of data encapsulates useful signals driven by our everyday life and therefore, statistical learning methods could be applied to extract them (several examples are provided in Section 2).

The term *nowcasting*, commonly used in finance, expresses the fact that we are making inferences regarding the current magnitude $\mathcal{M}(\varepsilon)$ of an event ε. For a time interval $u = [t - \Delta t, t]$, where t denotes the current time instance, consider $\mathcal{M}\left(\varepsilon^{(u)}\right)$ as a latent variable. The Web content $\mathcal{W}^{(u)}$ for this time interval is a partially observed variable; in particular, data from a social network, denoted as $\mathcal{S}^{(u)} \subseteq \mathcal{W}^{(u)}$ are being observed. In this work, $\mathcal{S}^{(u)}$ is used to directly infer $\mathcal{M}\left(\varepsilon^{(u)}\right)$. For short time intervals u, we are inferring the present value of the latent variable, that is, we are nowcasting the magnitude of an event. We have already presented preliminary results on a methodology for tracking the level of a flu epidemic from Twitter content using unigrams [Lampos and Cristianini 2010] and demonstrated an online tool[2] for this purpose [Lampos et al. 2010]. Here we extend our previous findings and present a general framework for exploiting user input published in social media.

Sparse learning enables us to select a consistent set of features (e.g., unigrams or bigrams) and then use it to perform inference via regression. The performance of the proposed methodology is evaluated by investigating two case studies. In the first, we infer the daily amount of rainfall in five UK locations by using tweets; this forms a benchmark problem testing the limits of our approach given that rainfall has a very inconsistent behavior in the UK [Jenkins et al. 2008]. Ground truth consists of rainfall observations taken from weather stations located in the vicinity of the target locations. The second case study focuses on inferring the level of Influenza-like Illness (ILI) in the population of three UK regions based again on geolocated Twitter content. Results are validated by being compared with actual ILI rates measured by the Health Protection Agency (HPA).[3] In both case studies, experimental results are very positive in terms of the semantic correlation between selected features and target topics, and strong given the general inference performance.

The specific procedure that we followed, namely using Bolasso [Bach 2008] for feature selection from a large set of candidates has proven to work best compared to another relevant state-of-the-art approach [Ginsberg et al. 2008], 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. Comparisons with other variants of Machine Learning methods may be 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., Corley et al. [2009]; Polgreen et al. [2008]) 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 (see Sections 5, 6, and 7), 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. 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 dataset of Twitter posts and actual HPA readings; similar

[2]Flu Detector, `http://geopatterns.enm.bris.ac.uk/epidemics/`.
[3]HPA's weekly epidemiological updates archive is available at `http://goo.gl/wJex`.

results are derived for the rainfall case study. 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 function of the signal in Twitter.

The content of this article is laid out as follows: related work and background theoretical foundations are provided in Section 2; the proposed methodology, the performance evaluation procedure and the baseline approach, to which we compare our results, are described in Section 3; Section 4 is concerned with the technical details of information collection and retrieval explaining how Twitter is sampled and also defines the classes of features used in our approach; Sections 5 and 6 include a detailed presentation and analysis of the experimental results for the case studies of rainfall and flu nowcasting respectively; finally, Section 7 further discusses the derivations of this work, followed by the conclusions and future work in Section 8.

2. RELATED WORK AND THEORETICAL FOUNDATIONS

2.1. Related Work in Mining User-Generated Content

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, for instance, flu related keywords, or the application of sentiment/mood analysis, which in turn implies the use of predefined vocabularies, where words or phrases have been mapped to sentiment or mood scores [Pang and Lee 2008]. Corley et al. [2009] reported a 76.7% correlation between official ILI rates and the frequency of certain hand-picked influenza related words in blog posts [Corley et al. 2009], whereas similar correlations were shown between user search queries that included illness related words and CDC[4] rates [Polgreen et al. 2008]. Furthermore, sentiment analysis has been applied in the effort of extracting voting intentions [Tumasjan et al. 2010] or box-office revenues [Asur and Huberman 2010] from Twitter content. Similarly, mood analysis combined with a nonlinear regression model derived an 87.6% correlation with daily changes in Dow Jones Industrial Average closing values [Bollen et al. 2011]. Finally, Sakaki et al. [2010] presented a method that exploited the content, time stamp and location of a tweet to detect the existence of an earthquake.

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 [Guyon and Elisseeff 2003]. In Ginsberg et al. [2008] 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 90% 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 [Culotta 2010]. 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, which applied sparse regression on Twitter content for automatic feature selection, resulted to a greater than 90% correlation

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

with HPA's flu rates for several UK regions [Lampos and Cristianini 2010]; an improved version of this methodology has been incorporated in Flu Detector [Lampos et al. 2010], an online tool for inferring flu rates based on tweets.

Besides minor differences regarding the information extraction and retrieval techniques or the datasets considered, the fundamental distinction between Ginsberg et al. [2008] and Culotta [2010] and Lampos and Cristianini [2010] 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 that satisfies its constraints [Tibshirani 1996] (see Section 2.2). In this work, we extend and generalize the methodology and preliminary results presented in Lampos and Cristianini [2010]. The main theoretical concept is again feature selection by sparse learning, though we aim to make this selection consistent considering, at the same time, more types of features.

2.2. Bootstrapped LASSO for Feature Selection

Least Absolute Shrinkage and Selection Operator (LASSO), presented in Tibshirani [1996], being a constrained version of ordinary least squares (OLS) regression, provides a sparse regression estimate β^* computed by solving the following optimization problem:

$$\beta^* = \arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t, \tag{1}$$

where x's denote the input data (N observations of p variables), y's are the N target values, β's the N coefficients or weights, β_0 is the regression bias and $t \geq 0$ is referred to as the regularization or shrinkage parameter since it controls the regularization (or shrinkage) amount on the L1-norm of β's. Least Angle Regression (LARS) provides an efficient algorithm for computing the entire regularization path of LASSO [Efron et al. 2004], that is, all LASSO solutions for different choices of the regularization parameter t. However, it has been shown that LASSO selects more variables than necessary [Lv and Fan 2009] and that in many settings it performs an inconsistent model selection [Zhao and Yu 2006].

Bootstrap, presented in Efron [1979], was introduced as a method for assessing the accuracy of a prediction but has also found applications in improving the prediction itself (see for example *Bagging* [Breiman 1996]). Suppose that we aim to fit a model to a training dataset $\mathcal{T}$. The basic idea of bootstrapping is to draw n random datasets $\mathcal{B}$ with replacement from $\mathcal{T}$, forcing each sample to have the same size as $|\mathcal{T}|$; the drawn datasets are referred to as bootstraps. Then, refit the model into each element of $\mathcal{B}$ and examine the behavior of the fits [Efron and Tibshirani 1993]. The bootstrapped version of LASSO, conventionally named as *Bolasso*, intersects the supports of LASSO bootstrap estimates and addresses its model selection inconsistency problems [Bach 2008]. Throughout this work we have applied Bolasso's soft version (see Sections 3.1 and 3.2) in our effort to select a consistent subset of textual features.

3. GENERAL METHODOLOGY

In this section, a general description of the proposed methodology is given, introducing the notation that is going to be used throughout this script. An abstract summary of the methodology includes the following three main operations.

(1) *Candidate Feature Extraction*. A vocabulary of candidate features is formed by using n-grams, that is, 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, for instance, 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.

(2) *Vector Space Representation*. For a fixed time period and set of locations, the Vector Space Representation (VSR) of the candidate features is computed from the text corpus using a scheme based on Term Frequencies (TF). For the same time period and locations, the VSR of the target topic is obtained from an authoritative source.

(3) *Feature Selection and Inference*. A subset of the candidate features is selected by applying a sparse regression method. In our experiments, we have applied Bolasso to select a consistent set of features; the weights of the selected features are then learnt via OLS regression on the reduced input space. The selected features and their weights are used to perform inferences.

3.1. Formal Description

We denote the set of candidate n-grams as $\mathcal{C} = \{c_i\}$, $i \in \{1, ..., |\mathcal{C}|\}$. The retrieved user posts (or tweets) for a time instance u are denoted as $\mathcal{P}^{(u)} = \{p_j\}$, $j \in \{1, ..., |\mathcal{P}^{(u)}|\}$. A boolean function g indicates whether a candidate marker c_i is contained in a user post p_j or not:

$$g(c_i, p_j) = \begin{cases} 1 & \text{if } c_i \in p_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

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

$$s\left(c_i, \mathcal{P}^{(u)}\right) = \frac{\sum_{j=1}^{|\mathcal{P}^{(u)}|} g(c_i, p_j)}{|\mathcal{P}^{(u)}|}. \tag{3}$$

Therefore, the score of a candidate marker is the number of tweets containing this marker divided by the total number of tweets for a predefined time interval. The scores of all candidate markers for the same time interval u are kept in vector x given by:

$$x^{(u)} = \left[s\left(c_1, \mathcal{P}^{(u)}\right) \ ... \ s\left(c_{|\mathcal{C}|}, \mathcal{P}^{(u)}\right)\right]^T. \tag{4}$$

In our study, u takes the length of a day d; from this point onwards, consider a time interval equal to the duration of a day. However, u's length is a matter of choice depending on the inference task at hand.

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

$$\mathcal{X}^{(\mathcal{D})} = \left[x^{(d_1)} \ ... \ x^{(d_{|\mathcal{D}|})}\right]^T. \tag{5}$$

For the same set of $|\mathcal{D}|$ days, we retrieve the values of the target variable $y^{(\mathcal{D})}$:

$$y^{(\mathcal{D})} = \left[y_1 \ ... \ y_{|\mathcal{D}|}\right]^T. \tag{6}$$

$\mathcal{X}^{(\mathcal{D})}$ and $y^{(\mathcal{D})}$ are used as an input in Bolasso. 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:

$$\begin{aligned} &\min_{w} \ \left\| \mathcal{X}^{(\mathcal{D})} w - y^{(\mathcal{D})} \right\|_2^2 \\ &\text{s.t.} \qquad \|w\|_1 \leq t, \end{aligned} \tag{7}$$

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

$$t = \alpha \cdot \|w_{\text{OLS}}\|_1, \ \alpha \in (0, 1], \tag{8}$$

where w_{OLS} is the OLS regression solution and a denotes the desired shrinkage percentage of w_{OLS}'s L1-norm. Bolasso's implementation applies LARS, which is able to explore the entire regularization path at the cost of one matrix inversion and decides the value of the regularization parameter (t or α) using the largest consistent region, that is, the largest continuous range on the regularization path, where the set of selected variables remains the same [Efron et al. 2004].

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

$$\min_{w_s} \ \left\| \left(\mathcal{Z}^{(\mathcal{D})} w_s + \beta \right) - y^{(\mathcal{D})} \right\|_2^2, \tag{9}$$

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

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 1-norm of w. For example in Bartlett et al. [2009] it is shown that LASSO's expected loss $\mathcal{L}(w)$ up to polylogarithmic factors in W_1, $|\mathcal{C}|$ and $|\mathcal{D}|$ is bounded by

$$\mathcal{L}(w) \leq \hat{\mathcal{L}}(w) + \mathcal{Q}, \text{ with } \mathcal{Q} \sim \min \left\{ \frac{W_1^2}{|\mathcal{D}|} + \frac{|\mathcal{C}|}{|\mathcal{D}|}, \frac{W_1^2}{|\mathcal{D}|} + \frac{W_1}{\sqrt{|\mathcal{D}|}} \right\}, \tag{10}$$

where $\hat{\mathcal{L}}(w)$ denotes the empirical loss, $|\mathcal{C}|$ is the number of candidate features, $|\mathcal{D}|$ is the number of training samples and W_1 is an upper bound for the 1-norm of w, that is, $\|w\|_1 \leq W_1$. Therefore, to minimize 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 ($|\mathcal{C}|$) or increase the shrinkage of w's 1-norm (which intuitively might result in sparser solutions).

3.2. Consensus Threshold and Performance Evaluation

A strict application of Bolasso implies that only features with a nonzero weight in all bootstraps are going to be considered. In our methodology a soft version of Bolasso is applied (named as *Bolasso-S* in Bach [2008]), where features are considered if they acquire a nonzero 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, performance evaluation 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.

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

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

$$\mathcal{L}(w) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \ell\left(\langle x_i w\rangle, y_i\right), \tag{11}$$

where the loss function $\ell(\langle x_i w\rangle, y_i) = (\langle x_i w\rangle - y_i)^2$. The Root Mean Squared Error (RMSE) – the square root of MSE – has been used as a more comprehensive metric (it has the same units with the target variables) for presenting results in Sections 5 and 6.

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, it is given by:

$$i^{(*)} = \arg\min_i \mathcal{L}_i^{(\text{val})}. \tag{12}$$

Therefore, $\text{CT}_{i^{(*)}}$ 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, that is, $x_i \leftarrow \max\{x_i, 0\}$. 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.

As part of the evaluation process, we compare our results with a baseline approach that encapsulates the methodologies in Ginsberg et al. [2008] and Culotta [2010]. Those approaches, as explained in Section 2, mainly differ in the feature selection process that is performed via correlation analysis (Algorithm 1). 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.

4. DATA COLLECTION AND INFORMATION RETRIEVAL

For the experimental purposes of this work we use millions of tweets collected via Twitter's Search API and ground truth from authoritative sources. Based on the fact that information geolocation is a key concept in both case studies, we are considering only tweets tagged with the location (longitude and latitude coordinates) of their author. We use UK's 54 most populated urban centers and collect tweets geolocated within a 10km range from each one of them. Our crawler exploits *Atom* feeds and periodically retrieves the 100 most recent tweets per urban center.[5] The time interval between consecutive queries for an urban center varied from 5 to 10 minutes but has been stable on a daily basis and always the same for all locations. Therefore, a sampling method is carried out during collection; we try to reduce sampling biases (for the purposes of our work) by using the same sampling frequency per urban center. Collecting all tweets, apart from being a much more resource demanding process, would also have resulted in exceeding data collection limits set by Twitter. Nonetheless, the daily number of collected tweets (more than 200,000) is considered adequate for the experimental part of this work.

All collected tweets are stored and indexed in a MySQL database. Text preprocessing such as stemming by applying Porter's Algorithm for English language [Porter 1980], stop word and punctuation removal as well as the computation of VSRs are performed by our software libraries. VSRs are formed using a TF binary vector space model as already described in Section 3.1.

Candidate features, that is, the pool or vocabulary of n-grams on which feature selection is applied, are extracted from encyclopedic, scientific, or more informal Web references related to the inference topic.[6] By performing feature extraction in this way, we secure the existence of good candidate features, but we are also enabled to test the feature selection capability of our method, since most candidates are not directly related to the target topic. A typical information retrieval approach would have implied the creation of a vocabulary index from the entire Twitter corpus [Manning et al. 2008]; our choice is extensively justified in the discussion section. Nevertheless, acquired results indicate that our simplification in the feature extraction process results in a significant inference performance.

4.1. 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 characterized 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), their occurrences are expected to be sometimes close to zero.

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. Validation and testing are performed on the combined datasets. Suppose that for all considered consensus thresholds CT_i, $i \in \{1, ..., |CT|\}$, 1-grams

[5]For an area formed by a center with coordinates –latitude and longitude– (X,Y) and a radius of R Km, the N most recent tweets written in English language are retrieved by performing the following query: `http://search.twitter.com/search.atom?geocode=X,Y,R&lang=en&rpp=N`.

[6]Lists of Web references and extracted features for the investigated case studies in this article are available at `http://geopatterns.enm.bris.ac.uk/twitter/`.

and 2-grams selected via Bolasso are denoted by $\mathcal{F}_i^{(U)}$ and $\mathcal{F}_i^{(B)}$ respectively. Then, the pseudo-selected n-grams for all CTs for the hybrid class $\mathcal{F}_i^{(H)}$ are formed by their union, $\mathcal{F}_i^{(H)} = \left\{\mathcal{F}_i^{(U)} \cup \mathcal{F}_i^{(B)}\right\}, i \in \{1, ..., |CT|\}$. Likewise, $\mathcal{Z}_i^{(H)} = \left\{\mathcal{Z}_i^{(U)} \cup \mathcal{Z}_i^{(B)}\right\}, i \in \{1, ..., |CT|\}$, where $\mathcal{Z}$ denotes the VSR of each feature class (using Section's 3.1 notation). Validation and testing are performed on $\mathcal{Z}_i^{(H)}$ as it has already been described in Section 3.2. 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 (see Sections 5 and 6) do indeed indicate that feature class H performs on average better than U and B.

5. 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, that is, 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 allowing 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 nonsmooth behavior [Jenkins et al. 2008].

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. As already mentioned in Section 4, the majority of the extracted candidate features is not directly related to the target topic, but there exists a subset of markers that 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.

5.1. Experimental Settings

A year of Twitter data and rainfall observations (from the July 1, 2009 to the June 30, 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 Bach [2008]), 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 that guarantees a quicker execution of the learning phase, especially when dealing with large amounts of data.

The performance of each feature class 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 their weights. Training is performed by using the VSRs of all five locations in

Table I.

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*)

Table II.

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

a batch dataset, CT's validation is carried out on the same principle (i.e., we learn the same markers-weights under the same CT for all locations), and finally testing is done both on the batch dataset (to retrieve a total performance evaluation) and on each location separately. Finally, we also compute the inference performance of the baseline approach for feature selection (Algorithm 1) for the same training, validation, and testing sets, considering the top $k = 300$ correlated terms.

5.2. 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 I. In most rounds CT values are close to 90% meaning that a few markers were able to capture the rainfall rates signal. However, in the last round, where the validation dataset is based on July 2009, CTs for all feature classes are significantly lower, which can be interpreted by the fact that July is the 2nd most rainy month in our dataset, but also a summer month; therefore, tweets for rain could be followed or preceded by tweets discussing a sunny day, creating instabilities during the validation process. In addition, our dataset is restricted to only one year of weather observations, and therefore seasonal patterns like this one are not expected to be captured properly.

Table III.

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 IV.

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 V.

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		

Detailed performance evaluation results (total and per location for all feature classes) are presented on Table II. For a better interpretation of the numerical values (in mm), consider that the average rainfall rate in our dataset 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 1 in Section 3.2) in all-but-one intermediate RMSE indications as well as in total for all feature classes, 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 both U and B – in the results per location, feature class H has the best performance 11 times (out of 30), the same holds for B, and U is better 8 times.

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 III, IV, and V list the selected features in alphabetical order together with their weights for feature classes U, B and H respectively. For class H we have also compiled a word cloud with the selected features as a more comprehensive representation of the selection outcome (Figure 1). The majority of the selected 1-grams (Table III) 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 can act as mitigators of non weather related uses of the remaining rainy weather oriented and positively weighted features. The selected 2-grams (Table IV) have a clearer semantic connection with the topic with 'pour rain' acquiring the highest weight. In this particular case, the features for class H are

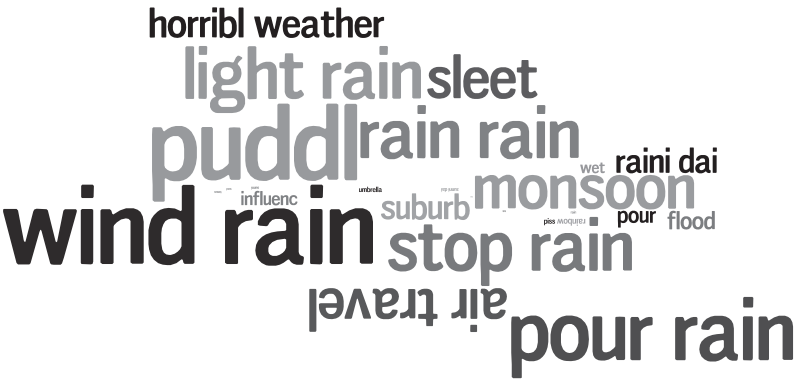
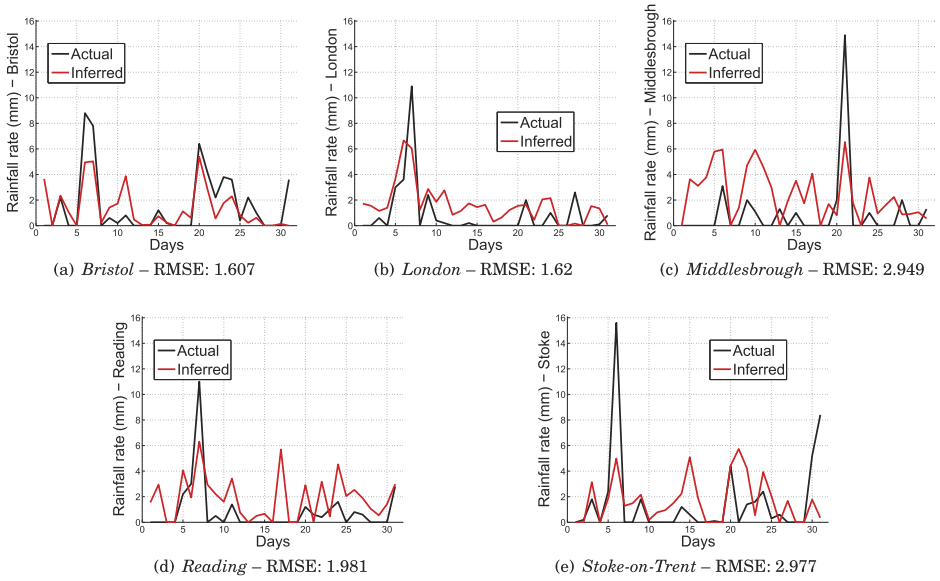


Fig. 1. Table V in a word cloud, where font size is proportional to regression's weight and flipped words have negative weights.



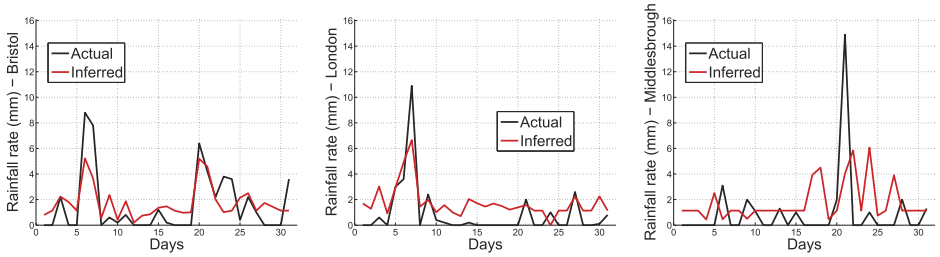
(a) *Bristol* – RMSE: 1.607 (b) *London* – RMSE: 1.62 (c) *Middlesbrough* – RMSE: 2.949

(d) *Reading* – RMSE: 1.981 (e) *Stoke-on-Trent* – RMSE: 2.977

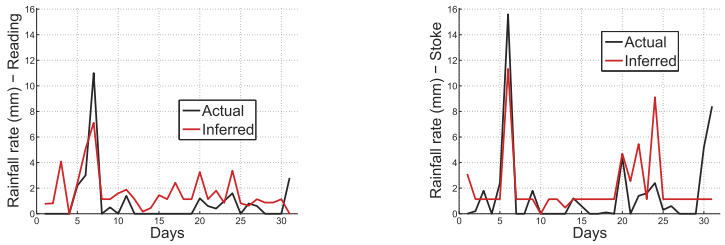
Fig. 2. Feature Class U – Inference for Rainfall case study (Round 5 of 6-fold cross validation).

formed by the exact union of the ones in classes U and B, but take different weights (Table V and Figure 1).

Inference results per location for cross validation's round 5 are presented in Figures 2, 3, and 4 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 3(e)) that 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 4(a) (Bristol), 4(b) (London) and 4(d) (Reading) demonstrate a good fit with the target signal.

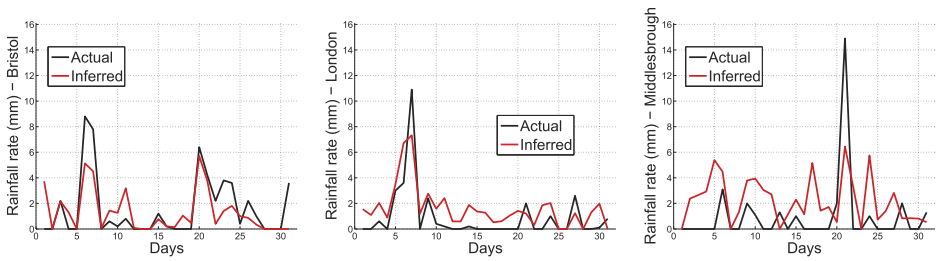


(a) *Bristol* – RMSE: 1.59 (b) *London* – RMSE: 1.478 (c) *Middlesbrough* – RMSE: 2.918

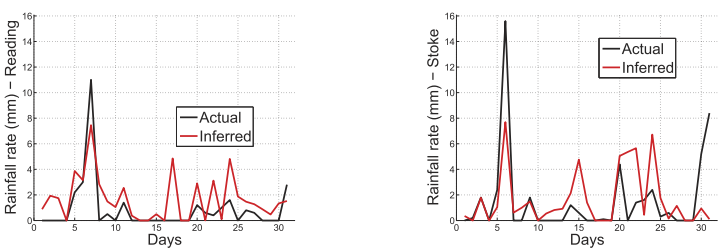


(d) *Reading* – RMSE: 1.572 (e) *Stoke-on-Trent* – RMSE: 2.447

Fig. 3. 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

Fig. 4. Feature Class H – Inference for Rainfall case study (Round 5 of 6-fold cross validation).

6. 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 and Wales, North England and South England. Ground truth, that is, official flu rate measurements, is derived from HPA. HPA's weekly reports are based on information collected from the Royal College of General Practitioners (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 100 are normal, between 100 and 200 are above average and over 200 are characterized as exceptional.[7] 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 2 consequent weeks, we compute a step factor δ from

$$\delta = \frac{r_{i+1} - r_i}{7}, \tag{13}$$

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

$$d_j = d_{j-1} + \delta,\ j \in \{2, ..., 7\}, \tag{14}$$

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, that is, 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.

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 nonflu 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 10(c); we apply the same randomized index on all regions during performance evaluation.

6.1. Experimental Settings

For this experiment, we considered tweets and ground truth in the time period between the June 21, 2009 and April 19, 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.

6.2. Results

The derived CTs and numbers of selected features for all rounds of the 5-fold cross validation are presented on Table VI. In the flu case study, most CTs (especially in U and H feature classes) get a value close to the lower bound (50%) after validation, and

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

Table VI.

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*)

Table VII.

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	U	11.781	9.005	16.147	13.252	10.912	12.219	12.677
& *Wales*	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

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 VII 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 29.270 and ranges in [2, 172]. Again, feature class U performs better than B, whereas H outperforms U and B. In the regional results per fold, class H has the best performance 8 times, B 5 times and U only 2 times (out of 15 subcases in total). 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 VIII, IX, and X show the selected features for U, B and H feature classes respectively. From the selected 1-grams (Table VIII), stem 'irrig'[8] 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 IX) can be considered as flu-related; 'confirm swine' has the largest weight for both feature classes B and H (Table X and Figure 5). As a general remark, keeping in mind that

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

Table VIII.

Feature Class U – 1-grams selected by Bolasso for Flu case study (Round 1 of 5-fold cross validation) – All weights (**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 IX.

Feature Class B – 2-grams selected by Bolasso for Flu case study (Round 1 of 5-fold cross validation) – All weights (**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		

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 6, 7, and 8 for classes U, B, and H respectively. There is a clear indication that the inferred signal has a strong correlation with the actual one; for instance, for feature class H (Figure 8) 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 and 0.911 respectively, providing additional evidence for the significance of the inference performance.[9]

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

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

Table X.

Feature Class H – Hybrid selection of 1-grams and 2-grams for Flu case study (Round 1 of 5-fold cross validation) – All weights (**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

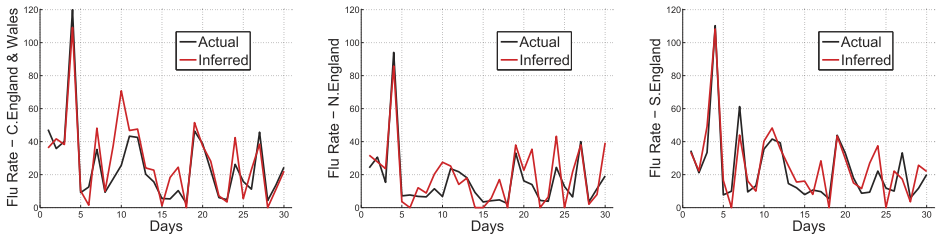
days of data, we used days 61–90 for validating CT, 91–121 for testing (from September 19 to October 19, 2009) and the remaining days have been used for training. In this formation setting, we train on data from the Swine Flu epidemic period and then test on a period where influenza existed but its rate was within a normal range. In Figure 9, we show the inference outcome for South England for all feature classes.[10] 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.

7. 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

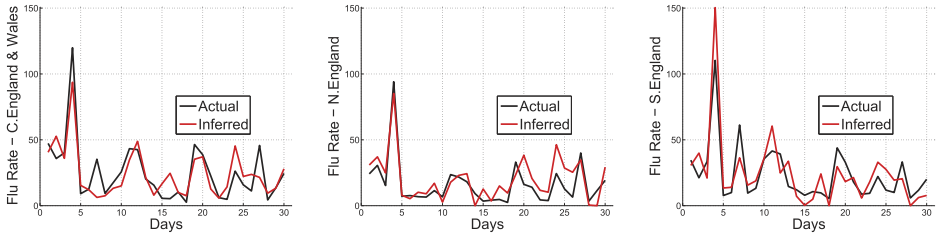
[10]Results for the other 2 regions were similar. South England region was chosen because it is the one with the highest population (as it includes the city of London).

Fig. 5. Table X in a word cloud, where font size is proportional to regression's weight and flipped words have negative weights.



(a) *C. England & Wales* – RMSE: 11.781 (b) *N. England* – RMSE: 9.757 (c) *S. England* – RMSE: 9.599

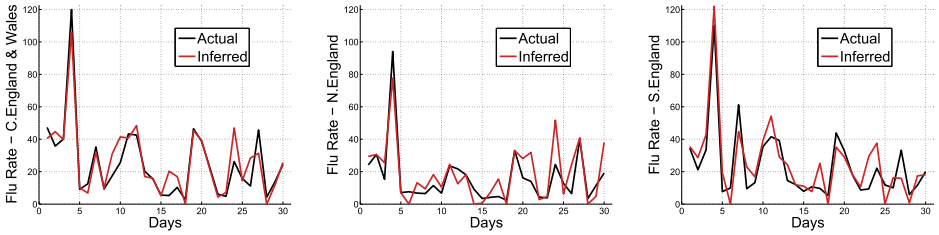
Fig. 6. Feature Class U – Inference for Flu case study (Round 1 of 5-fold cross validation).



(a) *C. England & Wales* – RMSE: 11.901 (b) *N. England* – RMSE: 9.659 (c) *S. England* – RMSE: 13.536

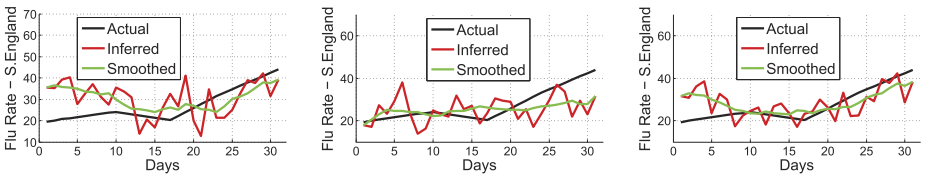
Fig. 7. Feature Class B – Inference for Flu case study (Round 1 of 5-fold cross validation).

discussions in the social Web media. While samples from both rainfall and ILI rates can be described by exponentially distributed random variables, those two phenomena have a distinctive property. Precipitation, especially in the UK, is rather unstable, that is, prone to daily changes, whereas a flu rate evolves much more smoothly. Figures 10(a) (rainfall in London) and 10(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 6th round of the cross-validation, where we derived the worst inference performance, we see that in the VSRs of the test set



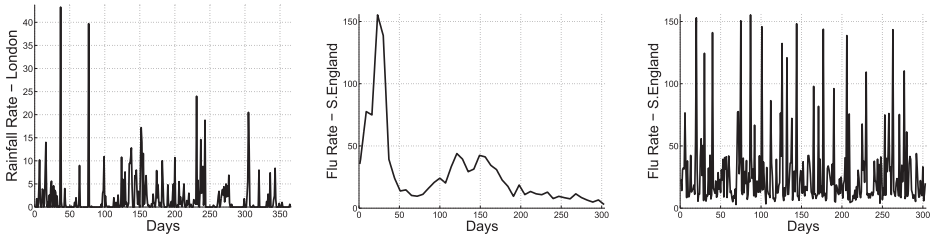
(a) *C. England & Wales* – RMSE: 8.36 (b) *N. England* – RMSE: 9.782 (c) *S. England* – RMSE: 9.86

Fig. 8. Feature Class H – Inference for Flu case study (Round 1 of 5-fold cross validation).



(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

Fig. 9. Flu inference results for continuous training, validating and testing sets for *South England* – Testing is performed on data from the 19th of September to the 19th October, 2009.



(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

Fig. 10. Comparing smoothness of ground truth between the two case studies.

(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; for instance, the average frequencies of 'rain hard' and 'pour rain' in the training set (716/1515 rainy days) for nonrainy 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 behavior 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 dataset, 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 that reach several extremes, we argue that the proposed method is applicable to other events as well, which are at least drawn from an exponential distribution.

Another important point in this methodology regards the feature extraction approach. A mainstream information retrieval technique implies the formation of a vocabulary index from the entire corpus [Manning et al. 2008]. 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 (see Equation (10)). 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 minimize the risk error and avoid overfitting. Indeed, in the flu case study, where only a small variation in the ILI rates is observed, when we formed an index from the entire Twitter corpus, the method tended to select non-illness-related features as well. Some of those features, for example, were describing a popular movie released in July 2009, the same period with the peak in the flu rates signal. By having fewer and slightly more focused on the target event's domain candidates, we constrain the dimensionality over training samples ratio and this issue is 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 that 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 dataset. 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 once in 22 validation sets the optimal value for CT was set equal to 1.

8. CONCLUSIONS AND FUTURE WORK

We have presented a supervised learning framework for nowcasting events by exploiting unstructured textual information published on the social Web. The proposed methodology is able to turn geo-tagged user posts on the microblogging service of Twitter to topic-specific geolocated signals by selecting textual features that capture semantic notions of the inference target. Sparse learning via a soft version of Bolasso, the bootstrapped LASSO L1-norm regulariser, performs a consistent feature selection, which increases the inference performance approx. by a factor of 10% compared to previously proposed methods [Culotta 2010; Ginsberg et al. 2008].

We have displayed results drawn from two case studies, that is, the benchmark problem of inferring rainfall rates and the real-life task of detecting the diffusion of Influenza-like Illness from tweets. In both case studies, the majority of selected features was directly related with the target topic and inference performance has been significant; for instance, for the important task of nowcasting influenza, inferred flu rates reached an average correlation of 91.11% with the actual ones. As expected, selected 2-grams showed a better semantic connection with the target topics. However, during inference they did not perform as well as 1-grams. Combining both feature classes into a hybrid approach resulted in an overall better performance.

Future work could be focused on improving various subtasks in our methodology. Feature extraction can become more sophisticated by identifying self diagnostic statements in the corpus (e.g., "I got soaked today" or "I have a headache") or by incorporating entities (gender, names, brands, etc.). Similarly to other work (mentioned

in Section 2), using sentiment or mood analysis on the text can offer an additional dimension of input information. Exploiting the temporal behavior of an event combined with more sophisticated inference techniques able to model non linearities or even a generative approach, could also improve the inference performance as well as provide interesting insights (e.g., the identification of latent variables that influence the inference process). Finally, on a conceptual basis, the detection of multivariate signals, where target variables may be interdependent (e.g., electoral voting intentions), could form an interesting task for future research.

ACKNOWLEDGMENTS

The authors would like to thank Twitter Inc. and HPA for making their data publicly accessible, PASCAL2 Network for the continuous support, and Tijl De Bie for providing feedback in early stages of this work. We are also grateful to the anonymous reviewers for their constructive feedback.

REFERENCES

ASUR, S. AND HUBERMAN, B. A. 2010. Predicting the future with social media. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE, 492–499.

BACH, F. R. 2008. Bolasso: Model consistent Lasso estimation through the bootstrap. In *Proceedings of the 25th International Conference on Machine Learning*. 33–40.

BARTLETT, P. L., MENDELSON, S., AND NEEMAN, J. 2009. l1-regularized linear regression: Persistence and oracle inequalities. Tech. rep., UC-Berkeley.

BOLLEN, J., MAO, H., AND ZENG, X. 2011. Twitter mood predicts the stock market. *J. Comput. Sci*.

BREIMAN, L. 1996. Bagging predictors. *Mach. Learn. 24,* 2, 123–140.

CORLEY, C. D., MIKLER, A. R., SINGH, K. P., AND COOK, D. J. 2009. Monitoring influenza trends through mining social media. In *Proceedings of the International Conference on Bioinformatics and Computational Biology*. 340–346.

CULOTTA, A. 2010. Towards detecting influenza epidemics by analyzing Twitter messages. In *Proceedings of the KDD Workshop on Social Media Analytics*.

EFRON, B. 1979. Bootstrap methods: Another look at the jackknife. *Ann. Statist. 7,* 1, 1–26.

EFRON, B. AND TIBSHIRANI, R. J. 1993. *An Introduction to the Bootstrap*. Chapman & Hall.

EFRON, B., HASTIE, T., JOHNSTONE, I., AND TIBSHIRANI, R. 2004. Least angle regression. *Ann. Statist. 32,* 2, 407–451.

GINSBERG, J., MOHEBBI, M. H., PATEL, R. S., BRAMMER, L., SMOLINSKI, M. S., AND BRILLIANT, L. 2008. Detecting influenza epidemics using search engine query data. *Nature 457,* 7232, 1012–1014.

GUYON, I. AND ELISSEEFF, A. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Resear. 3,* 7–8, 1157–1182.

JENKINS, G. J., PERRY, M. C., AND PRIOR, M. J. 2008. *The Climate of the United Kingdom and Recent Trends*. Met Office, Hadley Centre, Exeter, UK.

LAMPOS, V. AND CRISTIANINI, N. 2010. Tracking the flu pandemic by monitoring the Social Web. In *Proceedings of the 2nd IAPR Workshop on Cognitive Information Processing*. IEEE Press, 411–416.

LAMPOS, V., DE BIE, T., AND CRISTIANINI, N. 2010. Flu detector—Tracking epidemics on Twitter. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer, 599–602.

LV, J. AND FAN, Y. 2009. A unified approach to model selection and sparse recovery using regularized least squares. *Ann. Statist. 37,* 6A, 3498–3528.

MANNING, C. D., RAGHAVAN, P., AND SCHÜTZE, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

PANG, B. AND LEE, L. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retriev. 2,* 1–2, 1–135.

POLGREEN, P. M., CHEN, Y., PENNOCK, D. M., NELSON, F. D., AND WEINSTEIN, R. A. 2008. Using internet searches for influenza surveillance. *Clinical Infectious Diseases 47,* 11, 1443–1448.

PORTER, M. F. 1980. An algorithm for suffix stripping. *Program 14,* 3, 130–137.

SAKAKI, T., OKAZAKI, M., AND MATSUO, Y. 2010. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*. 851–860.

TIBSHIRANI, R. 1996. Regression shrinkage and selection via the lasso. *J. Royal Statist. Soc. Series B (Methodological) 58,* 1, 267–288.

TUMASJAN, A., SPRENGER, T. O., SANDNER, P. G., AND WELPE, I. M. 2010. Predicting elections with Twitter: What 140 characters reveal about political sentiment. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*. 178–185.

ZHAO, P. AND YU, B. 2006. On model selection consistency of Lasso. *J. Mach. Learn. Resear. 7,* 11, 2541–2563.

Received April 2011; revised August 2011; accepted September 2011