Multi-Task Learning for Spatio-Temporal Event Forecasting

Liang Zhao∗
Virginia Tech
liangz8@vt.edu

Qian Sun∗
Arizona State University
qsun21@asu.edu

Feng Chen
University at Albany - SUNY
fchen5@albany.edu

Chang-Tien Lu
Virginia Tech
cltu@vt.edu

Jieping Ye
University of Michigan
jpye@umich.edu

Naren Ramakrishnan
Virginia Tech
naren@cs.vt.edu

ABSTRACT

Spatial event forecasting from social media is an important problem but encounters critical challenges, such as dynamic patterns of features (keywords) and geographic heterogeneity (e.g., spatial correlations, imbalanced samples, and different populations in different locations). Most existing approaches (e.g., LASSO regression, dynamic query expansion, and burst detection) are designed to address some of these challenges, but not all of them. This paper proposes a novel multi-task learning framework which aims to concurrently address all the challenges. Specifically, given a collection of locations (e.g., cities), we propose to build forecasting models for all locations simultaneously by extracting and utilizing appropriate shared information that effectively increases the sample size for each location, thus improving the forecasting performance. We combine both static features derived from a predefined vocabulary by domain experts and dynamic features generated from dynamic query expansion in a multi-task feature learning framework; we investigate different strategies to balance homogeneity and diversity between static and dynamic terms. Efficient algorithms based on Iterative Group Hard Thresholding are developed to achieve efficient and effective model training and prediction. Extensive experimental evaluations on Twitter data from four different countries in Latin America demonstrated the effectiveness of our proposed approach.

Categories and Subject Descriptors

1.2.7 [Natural Language Processing]: Discourse; Text analysis

Keywords

Event forecasting; Multi-task learning; LASSO; Dynamic query expansion; Hard thresholding

∗These two authors contributed equally to this work

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

KDD ’15, August 11 - 14, 2015, Sydney, NSW, Australia
Copyright ©2015 ACM. ISBN 978-1-4503-3664-2/15/08...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2783377.

1. INTRODUCTION

Microblogs such as Twitter and Weibo are experiencing an explosive level of growth. Millions of worldwide microblog users broadcast their daily observations on an enormous variety of topics, e.g., crime, sports, and politics.

This paper focuses on the problem of spatial event forecasting from microblogs, for events such as civil unrest, disease outbreaks, and crime hotspots. The basic idea is to search for subtle patterns in specific cities as indicators of ongoing or future events, where each pattern is defined as a burst of context features (keywords) relevant to a specific event. For instance, the expression of discontent about gas price increases could be a potential precursor to a protest about government policies.

There are three technical challenges in addressing this problem: 1) Dynamic features. The language used in microblogs is highly informal, ungrammatical, and dynamic. Most existing methods treat fixed keywords as features [24, 26]. However, the expression in tweets may dynamically evolve, which makes the use of fixed features and historical training data insufficient. For example, the most significant Twitter keyword for the Mexican protests in Aug 2012 was “#YoSoy132” (i.e., the hashtag of an organization protesting against electoral fraud), alluding to the protests against the Mexican presidential election, but “#CNTE” (i.e., a hashtag denoting the national teacher’s association of Mexico) has become the most popular term by the beginning of 2013 due to the movements against the Mexican education reform. Ideally an event forecasting system must combine judicious use of static (fixed) features but must be cognizant to subtle changes involving dynamic features. 2) Geographic heterogeneity. Different cities have different characteristics, such as population, weather (e.g., humidity, temperature), and administrative structures (e.g., capital cities versus non-capital cities). As a result, it is difficult to impute basal levels of occurrence uniformly. Considering civil unrest as an example, finding 1000 tweets mentioning the keyword “protest” is likely not a strong indicator of an upcoming civil unrest event if the city houses a population of a few million users but could be a strong signal for a city with a population of 10,000. At the same time, it is difficult to dynamically adjust such thresholds precisely due to the data sparsity problems in the latter case. 3) Scalability. The massive scale of microblogging data necessitates development of new, scalable forecasting methods.

In order to concurrently address all these technical challenges, this work presents a novel computational approach in the framework of multi-task learning (MTL) that com-
The rest of this paper is organized as follows. Section 2 reviews background and related work, and Section 3 introduces the problem setup. Section 4 presents our multi-task feature learning models, and Section 5 presents efficient algorithms based on IGHT. Experiments on real Twitter datasets are presented in Section 6, and the paper concludes with a summary of the research in Section 7.

2. RELATED WORK

Compared to traditional media, Twitter has the following significant characteristics: 1) Timeliness of messages: Unlike traditional media that take hours or days to publish, tweets can be posted instantly utilizing portable mobile devices; 2) Ubiquity of social sensors: Tweets reflect the public’s mood and trends, which could be the determinants of future social events; and 3) Availability of geo-information: Twitter users provide rich location information in profiles, texts, and geotags. As a social “sensor” which can identify emerging patterns in sentiments and opinions, the use of microblogs holds great promise for detection and forecasting of significant societal events.

The typical dichotomy to event detection or forecasting research is to classify them into whether they are supervised or unsupervised. The former consider a set of stationary terms whose distribution can be learned from historical data. Particularly, LASSO regression methods estimate a sparse predictive model based on a predefined set of keyword terms (vocabulary) for each city that predicts the probability of an ongoing event in this city in each predefined time interval (e.g., hourly or daily) [22]. Burst detection methods search for geographic regions (cities) where the aggregated counts of some predefined terms are abnormally high compared with the counts outside the cities. For example, Sakaki et al. consider spatiotemporal Kalman filtering, which is similar to space-time burst detection, to track the geographical trajectory of hot spots of tweets related to earthquakes [24]. Unsupervised methods, as the name indicates, consider a set of dynamic terms that could be different in different time intervals, and apply unsupervised learning techniques for event detection. Particularly, the dynamic query expansion method (DQE) iteratively expand a predefined set of seed terms (e.g., protest, strike, march) using the current tweets to identify and rank new terms that are relevant to ongoing events, then retain the top terms and tweets containing these terms for further modeling [34]. Clustering-based methods search for novel spatial clusters of documents or terms using predefined similarity metrics, such as cosine similarity and social similarity for documents [3], or auto-correlations [2] and co-occurrences [32] for terms.

Event detection: A large body of work focuses on the identification of ongoing events, including earthquakes [24], disease outbreaks [26], and other types of events [3, 19, 32, 17]. In general, they either use classification or clustering to extract tweets of interest and examine the spatial [24], temporal [25, 32], or spatiotemporal burstiness [19] of the extracted tweets. However, instead of forecasting events in the future, these approaches typically can only uncover them after their occurrence.

Event forecasting: Most research in this area focuses on temporal events and ignores the underlying geographical information, such as the forecasting of elections [21, 29], stock...
Multi-task learning: Multi-task learning (MTL) learns multiple related tasks simultaneously to improve generalization performance [10, 27]. Many MTL approaches have been proposed in the past [36]. In [12], Evgeniou et al. proposed the regularized MTL which constrained the models of all tasks to be close to each other. The task relatedness posed the regularized MTL which constrained the models to be close to each other. The task relatedness of all tasks to be close to each other. The task relatedness tried to promote features of similar tasks to be close to each other. The task relatedness was defined as the task similarity, which is a measure of the similarity between tasks.

For each location \( l \), and time \( t \), define the number of tweets containing any of the \( k \) dynamic keywords \( S^{(k)} \) as \( D_{l,t}^{(k)} \). Then, the DQE-based event forecasting can be formulated as a function \( Y_{l,t} = I(D_{l,t} > \gamma) \), that is, \( Y_{l,t} = 1 \) if \( D_{l,t} \) is larger than the threshold \( \gamma \); \( Y_{l,t} = 0 \), otherwise. The dynamic keywords are expanded and ranked from the seed query based on the tweets data \( C_l \), where the seed query \( S_0 \) is an initial set of few semantically coherent keywords that characterize the concept of the targeted domain. Specifically, the keyword expansion process is formulated as follows:

\[
P_{l} = F_{l}(B_{l}^{T} \cdot B_{l} + B_{l}^{T} \cdot R_{l} \cdot B_{l}) \cdot P_{0}
\]

where \( P_{0} \in \mathbb{R}^{|V| \times 1} \) is the initial weight vector of all the words in \( V \), \( P_{0}(i) = I(\text{V} = \text{S}_0) \), and \( V_{l} \) is the \( \text{rth} \) word. \( B_{l} \) is the adjacency matrix between tweets and words. \( R_{l} \in \mathbb{R}^{V \times V} \) is the tweet-replying matrix, i.e., \( R_{l}(i,j) = 1 \) means there is replying relationship between tweet \( i \) and tweet \( j \), \( R_{l}(i,j) = 0 \), otherwise. \( F_{l} \in \mathbb{R}^{V \times V} \) is the inverse document frequency (IDF) matrix of \( F_{l} \), which is a diagonal matrix such that \( F_{l}^{(i)} \) refers to the IDF of the word \( V_{l} \). \( P_{l} \in \mathbb{R}^{V \times 1} \) is the updated weight vector. Finally, the dynamic keyword set \( S_{l}^{(k)} \) is defined as the top \( k \) words with the largest weights according to \( P_{l} \).

There are three main challenges for using either of LASSO and DQE individually: (1) The LASSO model only uses a set of predefined fixed keywords, called “static features,” which may not capture the fast-evolving expressions in Twitter, thus it may be difficult to predict future events that are related to a small set of new keywords not included in the fixed keywords set. (2) The LASSO model trains an individual model for each location, but many small cities may have insufficient amount of information in the training set to build an accurate forecasting model. (3) DQE requires two types of thresholds, which are \( 1 \) \( k \), the number of dynamic keywords expanded from a seed query, and \( \gamma \), the least number of tweets, each of which contains any of dynamic keywords, to indicate the event occurrence. However, it is difficult to set these two thresholds based on domain experience. In the next section, we present a novel computational approach based on multi-task learning to address all these three challenges.

4. MODELS

As defined above, LASSO uses the “static feature” set \( K_{l,1} \), which is the count of predefined keywords in location \( l \) at time \( t \). DQE uses the “dynamic feature” set \( D_{l,1,k} \), which is the number of tweets containing top \( k \) dynamic keywords at location \( l \) at time \( t \). Because it is difficult to predefine an optimal \( k \), we propose to make use of multiple \( k \) values in the range of \([1,s]\) (here \( s \) is user-specified parameter); our experiments show that using a set of \( s = 20 \) values is a separate sparse linear regression model for each location \( l \):

\[
\arg\min_{w_l} \| w_l^T K_{l,1} - Y_{l,t} \|_2^2 + \rho_1 \| w_l \|_1,
\]

where the regularization parameter \( \rho_1 \) controls the sparsity, and \( w_l \in \mathbb{R}_d \times 1 \) is the vector of regression coefficients that need to be estimated. We need to estimate \( m \cdot d \) parameters in total for the \( m \) separate LASSO regression models.

DQE is a Twitter-oriented query expansion method to get dynamic keywords, which are then utilized for event detection or forecasting. Denote \( I(\cdot) \) as the indicator function. For each location \( l \) and time \( t \), define the number of tweets containing any of the \( k \) dynamic keywords \( S_{l}^{(k)} \) as \( D_{l,1,k} \). Then, the DQE-based event forecasting can be formulated as a function \( Y_{l,t} = I(D_{l,t} > \gamma) \), that is, \( Y_{l,t} = 1 \) if \( D_{l,t} \) is larger than the threshold \( \gamma \); \( Y_{l,t} = 0 \), otherwise. The dynamic keywords are expanded and ranked from the seed query based on the tweets data \( C_l \), where the seed query \( S_0 \) is an initial set of few semantically coherent keywords that characterize the concept of the targeted domain. Specifically, the keyword expansion process is formulated as follows:

\[
P_{l} = F_{l}(B_{l}^{T} \cdot B_{l} + B_{l}^{T} \cdot R_{l} \cdot B_{l}) \cdot P_{0}
\]

where \( P_{0} \in \mathbb{R}^{1 \times 1} \) is the initial weight vector of all the words in \( V \), \( P_{0}(i) = I(\text{V} = \text{S}_0) \), and \( V_{l} \) is the \( \text{rth} \) word. \( B_{l} \) is the adjacency matrix between tweets and words. \( R_{l} \in \mathbb{R}^{V \times V} \) is the tweet-replying matrix, i.e., \( R_{l}(i,j) = 1 \) means there is replying relationship between tweet \( i \) and tweet \( j \), \( R_{l}(i,j) = 0 \), otherwise. \( F_{l} \in \mathbb{R}^{V \times V} \) is the inverse document frequency (IDF) matrix of \( F_{l} \), which is a diagonal matrix such that \( F_{l}^{(i)} \) refers to the IDF of the word \( V_{l} \). \( P_{l} \in \mathbb{R}^{V \times 1} \) is the updated weight vector. Finally, the dynamic keyword set \( S_{l}^{(k)} \) is defined as the top \( k \) words with the largest weights according to \( P_{l} \).
sufficient), and then learn the optimal \( k \) automatically in the proposed multi-task learning framework. This results in \( D_{i,t} = \{D_{i,t,k}\}_{k=1}^{m}, D_{i,t} \in \mathbb{R}^{K \times 1} \), called the “dynamic feature” set for location \( l \) and time \( t \). We combine the information used in LASSO and DQE by forming a new data matrix \( \mathbf{X}_{i,t} = [K_{i,t}; D_{i,t}] \in \mathbb{R}^{d+n \times n_i,t} \). For notational simplicity, we will remove subscript \( t \) throughout the rest of this paper.

We aim to build \( m \) models \( \{w_i| i = 1, \ldots, m\} \) to predict the occurrence of events for the \( m \) locations. A simple approach is to learn these \( m \) models (tasks) independently, ignoring the task relatedness. However, such an approach does not consider the intrinsic relationships among cities, and the resulting models may not be accurate as some cities may not have sufficient information in the training set. To address this issue, we propose to build the forecasting models for all \( m \) cities simultaneously by extracting and utilizing appropriate shared information across tasks [36]. Figure 1 illustrates the proposed multi-task learning framework. Learning multiple related tasks simultaneously effectively increases the sample size for each city, since when we learn a model for a specific city, we use information from all other cities.

Intuitively, the events that occur at different cities around the same time may involve similar topics, thus the tweets from different cities may share many common keywords that are related to the events. This motivates us to explore multi-task feature learning (MTFL) models which constrain multiple related models to select a common set of features. Specifically, we explore three multi-task feature learning models:

- Regularized multi-task feature learning model,
- Constrained multi-task feature learning model I,
- Constrained multi-task feature learning model II.

Each of the three models formulates the multi-task learning problem by following a general paradigm, i.e., to minimize a penalized empirical loss:
\[
\min_W f(W) + \lambda g(W) \tag{1}
\]

or a constrained version:
\[
\min_W f(W) \quad \text{s.t.} \quad g(W) \leq l. \tag{2}
\]

where \( f(W) \) is the empirical loss on the training set; we use a smooth and convex loss function, e.g., the least squares and logistics loss. \( g(W) \) is the regularization term that encodes task relatedness, which is typically non-smooth or even non-convex. \( \lambda \) (or \( l \)) is a tuning parameter to balance the tradeoff between the loss and penalty.

Different regularization/constraint terms capture different types of task relatedness [12, 1, 16, 11]. In this paper, we adopt the least square loss, and characterize the model relatedness by restricting all models to select a common set of features. We detail the three models below.

### 4.1 Regularized MTFL model

The \( j \)-th element in model \( w_j \) indicates the importance of \( j \)-th feature for \( i \)-th task. In MTFL, we restrict all tasks to share a common set of top features, that is, the forecasting models for all cities are based on the same subset of features. This can be achieved by grouping the \( j \)-th elements of all tasks together and selecting the top groups. Specifically, we consider the \( m \) entries of the \( j \)-th row of the matrix \( W \) as a group and use the \( l_{2,1} \)-norm regularization to identify the top groups [5]. Thus, the \( j \)-th feature which corresponds to the \( j \)-th element in models is likely to be selected or not by all models simultaneously, achieving our desired goal.

Mathematically, we employ the following multi-task feature learning model:
\[
\min_W \sum_{i=1}^{m} \|w_i^T X_i - Y_i\|_F^2 + \rho_1 \|W\|_{2,1} + \rho_2 \|W\|_F^2, \tag{3}
\]

where the first term is the data fitting term for all tasks, \( \|W\|_{2,1} \) denotes the \( l_{2,1} \) norm of matrix \( W \) which encourages all tasks to select a common set of features, and it can be computed as the summation of \( l_2 \)-norm of each row in \( W \). The regularization parameter \( \rho_1 \) controls the sparsity.

We include a small multiple of the Frobenius-norm regularization, i.e., \( \|W\|_F^2 \), to enhance the robustness of the model. Problem (3) is a convex problem and can be solved by the FISTA algorithm [7].

### 4.2 Constrained MTFL model I

In the regularized MTFL model above, the model sparsity is controlled by the parameter \( \rho_1 \), which is less interpretable than the number of features selected. It is thus desired to develop a model which directly controls the number of features to be selected. To this end, we introduce a constraint in the model which ensures that a specific number of rows of \( W \) will be non-zero, i.e., we control the number of features included in the model. In particular, we consider the
The following constrained multi-task feature learning model:

\[
\min_W \sum_{i=1}^{m} \|w_i^T X_i - Y_i\|_p^2 + \rho_1\|W\|_F^2,
\]

\[\text{s.t. } \sum_j I(\|w_j\| > 0) \leq r. \tag{4}\]

Here \(w_j\) is the \(j\)-th row of \(W\) and \(I(\cdot)\) is the indicator function. The constraint in (4) ensures that the number of nonzero rows of \(W\) is no larger than \(r\), ensuring no more than \(r\) features will be selected. Note that the convexity property does not hold any more for Model (4). We will use the iterative group hard thresholding framework to solve (4). More details are provided in the next section.

Recall Problem (4), and denote \(f(W) = \sum_{i=1}^{m} \|w_i^T X_i - Y_i\|_p^2 + \rho_1\|W\|_F^2\). The key idea of IGHT is to first use the gradient information at the current iterate to provide the first-order approximation of the objective function, then apply the projection operators to ensure the next iterate satisfies the given constraints. Specifically, we use the combination of the linear approximation of the function \(f(W)\) at a given

4.3 Constrained MTFL model II

The constrained model above does not distinguish the static and dynamic features. Recall that the first \(d\) features correspond to the \(d\) static features, while the last \(s\) features correspond to the use of \(s\) dynamic features. The feature values thus have very different meanings. In general, \(d\) is much larger than \(s\). In our experiments, \(d\) is around 2000, while \(s\) is around 10 to 20. Thus, it is desired to restrict the number of features selected from these two groups separately. In addition, in the current DQE model, only one dynamic feature is used and a common threshold value is applied for all cities in the same country. It is thus natural to restrict the number of dynamic features selected (out of the total \(s\) candidates) to be one. To achieve these goals, we propose the following model, which selects \(u\) features from the \(d\) static features, and selects \(v\) features from the \(s\) dynamic features:

\[
\min_W \sum_{i=1}^{m} \|w_i^T X_i - Y_i\|_p^2 + \rho_1\|W\|_F^2,
\]

\[\text{s.t. } \sum_j I(\|w_j\| > 0) \leq u,
\]

\[\sum_j I(\|w_j\| > 0) \leq v, \tag{5}\]

where \(W_K\) is the model matrix corresponding to the set of static features, and \(W_D\) is the model matrix corresponding to the set of dynamic features. We illustrate the structure of the model in Figure 2. Similar to Problem (4), \(u\) and \(v\) are user-specified parameters that control the number of features selected for the two sets of features, i.e., static feature set and dynamic feature set, respectively. We set \(v = 1\) in our experiments, however, our model is more general in that the user can select an arbitrary number of dynamic features.

Problem (5) is non-convex due to the use of nonconvex constraints. Similar to Problem (4), we will apply the Iterative Group Hard Thresholding algorithm to solve Problem (5). We show the details of our proposed algorithm for Problem (5) in the next section.

5. ALGORITHM

The FISTA algorithm performs well for convex problems [7, 36, 11]. However, both Problem (4) and Problem (5) are non-convex. Even worse, they both involve discrete constraints, which make the problems challenging to solve. Motivated by the success of the iterative hard thresholding algorithm for solving \(l_0\)-regularized problems [8] and the recent advances on nonconvex iterative shrinkage algorithm [14, 33], we propose to employ the Iterative Group Hard Thresholding framework to solve both problems. Note that Problem (4) is a special case of Problem (5) with \(v = 0\). We thus focus on Problem (5) only in the following discussion. The details are summarized in Algorithm 1.

Algorithm 1 The Proposed Algorithm

Require: \(X, Y, \rho, \eta > 1\)
Ensure: solution \(W\)
1: Initialize \(W^0, \alpha^0 \leftarrow 1\).
2: for \(i \leftarrow 1, 2, \ldots\) do do
3: Initialize \(L\)
4: repeat
5: \(S^i \leftarrow W^i - \frac{1}{\eta} \nabla f(W^i)\)
6: \(W^i \leftarrow \text{proj}(S^i)\) (defined in Lemma 1)
7: \(L \leftarrow \eta L\)
8: until line search criterion is satisfied
9: if the objective stop criterion satisfied then
10: return \(W^i\)
11: end if
12: end for

Recall Problem (4), and denote \(f(W) = \sum_{i=1}^{m} \|w_i^T X_i - Y_i\|_p^2 + \rho_1\|W\|_F^2\). The key idea of IGHT is to first use the gradient information at the current iterate to provide the first-order approximation of the objective function, then apply the projection operators to ensure the next iterate satisfies the given constraints. Specifically, we use the combination of the linear approximation of the function \(f(W)\) at a given
point $W^0$ and a quadratic penalty term, and solve the following problem:

$$
\min_W f(W^0) + \langle \nabla f(W^0), W - W^0 \rangle + \frac{\rho}{2} \|W - W^0\|_F^2,
$$

subject to:

$$
\sum_j I(\|w^j_k\| > 0) \leq u,
$$

$$
\sum_j I(\|w^j_D\| > 0) \leq v,
$$

where $\rho$ is a positive constant that can be estimated by a line search scheme. By ignoring the constants and rearranging the terms in Problem (6), we obtain the following sub-problem:

$$
\min \frac{1}{2} \|W - S\|_F^2
$$

subject to:

$$
\sum_j I(\|w^j_k\| > 0) \leq u
$$

$$
\sum_j I(\|w^j_D\| > 0) \leq v,
$$

where $S = W^0 - \frac{1}{\rho} \nabla f(W^0)$. Problem (7) aims to find the optimal point satisfying the constraint set that is closest to a fixed point $S$. We call it an Euclidean projection problem, denoted as $\text{proj}(\cdot)$, even if the constraint set is not convex. The key of the IGHT framework is to solve the projection problem in (7). It is not hard to show that Problem (7) admits a closed-form solution as it can be decomposed into two independent problems, one for each block of features, as summarized in the following lemma.

**Lemma 1.** The projection Problem (7) admits a closed-form solution given below:

$$
w^j_k = \begin{cases} 
S^j_k, & \text{if } j \in \Omega_K \\
0, & \text{otherwise}
\end{cases}
$$

and

$$
w^j_D = \begin{cases} 
S^j_D, & \text{if } j \in \Omega_D \\
0, & \text{otherwise}
\end{cases}
$$

where $S_K$ consists of the first $d$ rows of $S$, $S^j_k$ is the $j$-th row of $S_K$, $S_D$ consists of the last $s$ rows of $S$, $S^j_D$ is the $j$-th row of $S_D$, $\Omega_K$ is the index subset of $\{1, 2, \cdots, d\}$ of size $u$, including all rows of $S_K$ that are among the top $u$ rows of $S_K$ in term of the length of the row vector, and $\Omega_D$ is the index subset of $\{1, 2, \cdots, s\}$ of size $v$, including all rows of $S_D$ that are among the top $v$ rows of $S_D$ in term of the length of the row vector.

One remaining issue is how to estimate the step size, which determines the amount of movement made along a given search direction. In this paper, we apply the well-known Lipschitz criterion to select the step size.

6. EXPERIMENTS

In this section, we evaluate the performance of the three multi-task learning formulations. First, we evaluate the effectiveness and efficiency of the methods on real data in comparison with baseline methods on multiple event forecasting tasks. Then, we study the parameter sensitivity of the methods. Finally, we provide several empirical case studies of civil unrest event forecasting to demonstrate the usefulness of these forecasting models. All the experiments were conducted on a 64-bit machine with Intel(R) core(TM) quad-core processor (i7CPU@3.40GHz) and 16.0GB memory.

### 6.1 Experiment Setup

The raw data was obtained by randomly sampling 10% (by volume) of the Twitter data from July 2012 to May 2013 in 4 countries in Latin America including Brazil, Paraguay, Mexico, and Venezuela, as shown in Table 1. Twitter data collection was partitioned into a sequence of date-interval subcollections. The Twitter data for the period from July 1, 2012 to December 31, 2012 was used for training while the second half of the period, from January 1, 2013 to May 31, 2013, was used for the performance evaluation. The locations of the tweets were geocoded by the geocoder in [22].

<table>
<thead>
<tr>
<th>Country</th>
<th>#Tweets (million)</th>
<th>News source</th>
<th>#Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>60</td>
<td>O Globo; O Estado de Sao Paulo; Ultima Hora; G1</td>
<td>Mexico</td>
</tr>
<tr>
<td>Paraguay</td>
<td>8</td>
<td>ABC Color; Ultima Hora; La Nacion</td>
<td>1217</td>
</tr>
<tr>
<td>Mexico</td>
<td>678</td>
<td>La Jornada; Reforma; Milenio</td>
<td>678</td>
</tr>
<tr>
<td>Venezuela</td>
<td>45</td>
<td>El Universal; El Nacional; Ultimas Noticias</td>
<td>563</td>
</tr>
</tbody>
</table>

In this experiment, two types of features were utilized. As introduced above, the first type of features is static features, which examines the relevance of tweets to fixed keywords. Specifically, they are defined as the daily counts of the keywords in the tweets. These keywords include 614 civil unrest related words (such as “protest” and “riot”), 192 phrases (such as “election fraud”), and country-specific actors (e.g., political parties and public figures). For each keyword, its translations in Spanish, Portuguese, and English are all included. The second type is dynamic features, which examines the volume of tweets containing dynamic keywords. Specifically, dynamic features are a set of counts, where each count is the number of daily tweets containing any of the top $k$ ($k \in [1, s]$) dynamic keywords. The dynamic keywords were extracted and ranked based on dynamic query expansion (DQE) [34], which utilizes both semantic and social relationship to expand real-time keywords from seed query, as introduced in Section 3. The seed query includes: “protest”, “march”, “movement”, “patriotic”, “manifest”, and their translations in Spanish and Portuguese. In this experiment, $s$ was set to 20. Thus we have 20 dynamic features.

In the experiment, given the day-by-day tweets data, the event forecasting task is to predict whether there is an event...
or not in the next day for a specific city. To perform this task, we created a training set and a testing set for each city, where each data sample is the daily tweet observation with the above-mentioned features. On the training set, we set the label for each data sample as “1” if there is an event on next day; and “0”, otherwise. Three standard performance metrics are used for comparison: precision, recall, and F-measure. The predicted events were structured as tuples of (date, city). A predicted event is matched to a GSR event if both the date and city attributes are matched; Otherwise, it is considered as a false forecasting.

The following methods are included for performance comparison: 1). LASSO [28]. For each city, three LASSO models are trained utilizing different sets of features: i). both static and dynamic features, and ii). only static features (denoted as LASSO-K). The regularization parameters of these models for different cities are set based on 10-fold cross validation. 2). DQEF-based event forecasting (DQEF). This model only considers the dynamic features, as introduced in Section 3. The number of top dynamic keywords, \( k \), and the tweet count threshold \( \gamma \) are set for each cities by 10-fold cross-validation on training set. 3). DQEF+LASSO. For each city, it first uses DQEF method to do forecasting. If there is no predicted event, i.e., \( Y_{t,l} = 0 \), the LASSO model using only static features will be employed for forecasting. 4). Regularized Multi-task Feature Learning Model (rMTFL). For each country, a rMTFL model is built where each task is the event forecasting for a city. This model utilizes three sets of features: i). Both static and dynamic features, ii). only static features (denoted as rMTFL-K), and iii). only dynamic features (denoted as rMTFL-D). The regularization parameters \( \rho_1 \) and \( \rho_2 \) are set based on 10-fold cross-validation. 5). Constrained multi-task feature learning model I (CMTFL-I). For each country, a model is built where each task is the event forecasting for a city. All the tasks share the same features, i.e., both the static and dynamic features. The feature number constraint \( r \) and the regularization parameter \( \rho_1 \) are set based on 10-fold cross-validation. 6). Constrained multi-task feature learning model II (CMTFL-II). For each country, a model is built where each task is the event forecasting for a city. All the tasks share the same features, i.e., the static and dynamic features. We use the 10-fold cross-validation to set the regularization parameter \( \rho_1 \), the numbers of static features \( u \), and dynamic features \( v \) for each country. The sensitivity of these three parameters are studied in Section 6.3.

### Table 2: Event forecasting performance comparison (Precision, Recall, F-measure)

<table>
<thead>
<tr>
<th></th>
<th>Mexico</th>
<th>Venezuela</th>
<th>Paraguay</th>
<th>Brazil</th>
<th>All Countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQEF+LASSO</td>
<td>0.57, 0.49, 0.53</td>
<td>0.59, 0.64, 0.61</td>
<td>1.00, 0.11, 0.20</td>
<td>0.42, 0.49, 0.45</td>
<td>0.55, 0.44, 0.49</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.70, 0.36, 0.48</td>
<td>0.94, 0.19, 0.32</td>
<td>1.00, 0.17, 0.29</td>
<td>0.63, 0.43, 0.51</td>
<td>0.73, 0.30, 0.43</td>
</tr>
<tr>
<td>rMTFL-D</td>
<td>0.96, 0.12, 0.21</td>
<td>0.66, 0.42, 0.51</td>
<td>1.00, 0.02, 0.04</td>
<td>1.00, 0.07, 0.13</td>
<td>0.77, 0.15, 0.25</td>
</tr>
<tr>
<td>rMTFL-K</td>
<td>0.78, 0.45, 0.57</td>
<td>0.53, 0.68, 0.60</td>
<td>0.93, 0.43, 0.59</td>
<td>0.79, 0.55, 0.65</td>
<td>0.71, 0.51, 0.59</td>
</tr>
<tr>
<td>rMTFL</td>
<td>0.70, 0.70, 0.70</td>
<td>0.54, 0.61, 0.57</td>
<td>0.71, 0.52, 0.66</td>
<td>0.68, 0.57, 0.62</td>
<td>0.96, 0.52, 0.68</td>
</tr>
<tr>
<td>CMTFL-I</td>
<td>0.59, 0.87, 0.70</td>
<td>0.51, 0.66, 0.58</td>
<td>0.95, 0.39, 0.55</td>
<td>0.72, 0.60, 0.66</td>
<td>0.62, 0.68, 0.65</td>
</tr>
<tr>
<td>CMTFL-II</td>
<td>0.71, 0.79, 0.75</td>
<td>0.53, 0.57, 0.55</td>
<td>0.78, 0.81, 0.79</td>
<td>0.76, 0.57, 0.65</td>
<td>0.69, 0.71, 0.70</td>
</tr>
</tbody>
</table>

### Table 3: Run time comparison of different methods

<table>
<thead>
<tr>
<th></th>
<th>Mexico</th>
<th>Venezuela</th>
<th>Paraguay</th>
<th>Brazil</th>
<th>All Countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training time (sec)</td>
<td>10.73</td>
<td>8.79</td>
<td>10.60</td>
<td>2.30</td>
<td>6.53</td>
</tr>
<tr>
<td>Testing time (sec)</td>
<td>0.003</td>
<td>0.001</td>
<td>0.003</td>
<td>0.01</td>
<td>0.001</td>
</tr>
</tbody>
</table>

### 6.2 Performance

Table 2 summarizes the comparison among the proposed methods and the competing methods for the task of civil unrest event forecasting. These results showed that the methods that utilize both sets of static features and dynamic features performed better than the ones utilizing either one of them. For example, rMTFL outperformed rMTFL-D and rMTFL-K by 50% and 10% in F-measure, respectively. DQEF+LASSO and LASSO outperformed LASSO-K by 10% on average in F-measure. All these results demonstrated effectiveness of combining both type of features for event forecasting. Among all methods, CMTFL-II achieved a recall of 0.71 and a F-measure of 0.70, which were both the best. Moreover, the proposed CMTFL-II performed well consistently across different countries by being the best in Mexico and Paraguay, and competitive in Venezuela and Brazil. Other methods like the proposed CMTFL-I and rMTFL also obtained high F-measures, around 0.65, but not as competitive as the CMTFL-II. The reason is because (1) CMTFL-II is able to ensure the inclusion of both type of features, whose combination are demonstrated to be more effective than using either one of them, and (2) unlike rMTFL and CMTFL-I, CMTFL-II treats both types of features separately in the constraint based on their different characteristics, leading to a more effective integration of these two types of features. Finally, we can observe from Table 2 that the multi-task models performed better than the traditional LASSO models by 50% on average. This revealed the advantage of multi-task models, which can select features by learning from similar forecasting tasks for all the cities. The generalization and stability of the forecasting performance can be improved by learning models for different cities together, especially for those cities that lack sufficient training samples.

Table 3 shows the run time of all the methods in training and testing. The training time of the multi-task models is only slightly larger than that of the LASSO model. As expected, the models using both static and dynamic features tend to consume more time than the ones only using either type of features. All methods consumed negligible testing time (around 0.01 0.003 sec).

Table 4 shows the specific features selected by different models, including rMTFL, LASSO, and the proposed CMTFL-II for several cities of two countries, i.e., Mexico (Spanish-spoken) and Brazil (Portuguese-spoken). According to Table 4, CMTFL-II effectively selected static features (i.e., keywords) very relevant to civil unrest, and the selection was...
Table 4: Top 10 static features (translated in English) and the selection of dynamic features. TRUE means there is at least one dynamic feature selected; FALSE means no dynamic feature selected. rMTFL and CMTFL-II can ensure sufficient and stable selection of static features. CMTFL-II can ensure the selection of effective dynamic feature(s).

<table>
<thead>
<tr>
<th>Methods</th>
<th>Features</th>
<th>Mexico City</th>
<th>Cuernavaca</th>
<th>Guadalajara</th>
<th>Morelia</th>
<th>Oaxaca</th>
<th>Brasilia</th>
<th>Rio de Janeiro</th>
<th>São Paulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>light</td>
<td>light</td>
<td>remember</td>
<td>employ</td>
<td>university</td>
<td>participant</td>
<td>expensive</td>
<td>prisoners</td>
<td></td>
</tr>
<tr>
<td></td>
<td>movement</td>
<td>hate</td>
<td>street</td>
<td>remember</td>
<td>allow</td>
<td>increased</td>
<td>strength</td>
<td>expensive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>election</td>
<td>hungry</td>
<td>work</td>
<td>unit</td>
<td>work</td>
<td>expensive</td>
<td>gringo</td>
<td>increase</td>
<td></td>
</tr>
<tr>
<td></td>
<td>president</td>
<td>street</td>
<td>hate</td>
<td>water</td>
<td>develop</td>
<td>prepare</td>
<td>cries</td>
<td>cries</td>
<td></td>
</tr>
<tr>
<td></td>
<td>congress</td>
<td>sent</td>
<td>president</td>
<td>university</td>
<td>hatred</td>
<td>include</td>
<td>protest</td>
<td>force</td>
<td></td>
</tr>
<tr>
<td></td>
<td>initiative</td>
<td>calling</td>
<td>unit</td>
<td>change</td>
<td>problem</td>
<td>protest</td>
<td>participant</td>
<td>include</td>
<td></td>
</tr>
<tr>
<td></td>
<td>progress</td>
<td>hungry</td>
<td>poor</td>
<td>class</td>
<td>progress</td>
<td>strength</td>
<td>protest</td>
<td>censorship</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hard</td>
<td>work</td>
<td>permit</td>
<td>statement</td>
<td>released</td>
<td>march</td>
<td>student</td>
<td>progress</td>
<td></td>
</tr>
<tr>
<td></td>
<td>help</td>
<td>eliminate</td>
<td>killing</td>
<td>force</td>
<td>congress</td>
<td>gringo</td>
<td>censorship</td>
<td>prepare</td>
<td></td>
</tr>
<tr>
<td></td>
<td>government</td>
<td>forcibly</td>
<td>remove</td>
<td>problem</td>
<td>killing</td>
<td>screams</td>
<td>include</td>
<td>student</td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>block</td>
<td>complaint</td>
<td>request</td>
<td>request</td>
<td>help</td>
<td>send</td>
<td>problem</td>
<td>throw</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fight</td>
<td>gunfire</td>
<td>confront</td>
<td>meet</td>
<td>power</td>
<td>power</td>
<td>water</td>
<td>bond</td>
<td></td>
</tr>
<tr>
<td></td>
<td>work</td>
<td>tranquility</td>
<td>water</td>
<td>water</td>
<td>avoid</td>
<td>food</td>
<td>official</td>
<td>unit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>help</td>
<td>forward</td>
<td>danger</td>
<td>danger</td>
<td></td>
<td>forward</td>
<td>work</td>
<td>defeat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hearsay</td>
<td>power</td>
<td>results</td>
<td>results</td>
<td></td>
<td>money</td>
<td>fight</td>
<td>send</td>
<td></td>
</tr>
<tr>
<td></td>
<td>president</td>
<td>avoid</td>
<td>order</td>
<td>order</td>
<td></td>
<td>street</td>
<td>government</td>
<td>forward</td>
<td></td>
</tr>
<tr>
<td></td>
<td>initiation</td>
<td>help</td>
<td>help</td>
<td>help</td>
<td></td>
<td>national</td>
<td>employ</td>
<td>control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>occupy</td>
<td>national</td>
<td>national</td>
<td>national</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>request</td>
<td>initiative</td>
<td>town</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td>protest</td>
<td>police</td>
<td>university</td>
<td>movement</td>
<td>block</td>
<td>shooting</td>
<td>attack</td>
<td>march</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fire</td>
<td>protest</td>
<td>expected</td>
<td>occupy</td>
<td>money</td>
<td>order</td>
<td>block</td>
<td>resolve</td>
<td></td>
</tr>
<tr>
<td></td>
<td>president</td>
<td>struggle</td>
<td>movement</td>
<td>encounter</td>
<td>encounter</td>
<td>movement</td>
<td>occupy</td>
<td>attack</td>
<td></td>
</tr>
<tr>
<td></td>
<td>government</td>
<td>patriot</td>
<td>manifest</td>
<td>hunger</td>
<td>memories</td>
<td>throw</td>
<td>arrest</td>
<td>warrant</td>
<td></td>
</tr>
<tr>
<td></td>
<td>movement</td>
<td>occupy</td>
<td>national</td>
<td>change</td>
<td></td>
<td>government</td>
<td>control</td>
<td>payment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>death</td>
<td>hunger</td>
<td>hate</td>
<td>change</td>
<td>police</td>
<td>submit</td>
<td>kill</td>
<td>poor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>poor</td>
<td>student</td>
<td>change</td>
<td>request</td>
<td>occupy</td>
<td>march</td>
<td>followers</td>
<td>claim</td>
<td></td>
</tr>
<tr>
<td></td>
<td>national</td>
<td>block</td>
<td>class</td>
<td>fear</td>
<td>steal</td>
<td>national</td>
<td>throw</td>
<td>block</td>
<td></td>
</tr>
<tr>
<td></td>
<td>expected</td>
<td>wait</td>
<td>memories</td>
<td>official</td>
<td>country</td>
<td>president</td>
<td>attack</td>
<td>march</td>
<td></td>
</tr>
<tr>
<td></td>
<td>wait</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

stable and consistent across different cities. Moreover, the selection of dynamic feature(s) was ensured, as shown in the bottom row, which enhanced the capacity to consider the burstiness of tweets containing dynamic keywords. rMTFL model also effectively selected civil unrest-related keywords as the top static features. However, it cannot guarantee the selection of dynamic features because in all the listed cities for Brazil, it did not select any dynamic features. The selected static features for LASSO model was not consistent across different cities, and more importantly, not as relevant and sufficient as the above-two multi-task learning models in several cities, especially the smaller ones, such as Oaxaca and Cuernavaca. Additionally, the selection of dynamic features was not ensured, such as in Morelia and Brasilia.

6.3 Parameter Sensitivity Study

There are three main parameters in the proposed rMTFL II model, which are the regularization parameter $\rho_1$, number of selected static features $u$, and number of selected dynamic features $v$.

Figure 3 illustrates the performance of the proposed model versus $\rho_1$, the regularization parameter. By varying $\rho_1$ in a large range from 0.001 to 500, the performance in F-measures for all the 4 countries are stable. The fluctuation ranges are typically within 8%.

Figure 4 shows the sensitivity results of varying $u$, the number of selected static features from 10 to 100. In general, for all the countries, the F-measures at $u = 10$ and $u = 20$ are slightly lower than other cases, but after $u$ is larger than 30, the F-measure becomes stable. This is because a

Figure 4: Sensitivity analysis on the number of selected static features.
Figure 5: Sensitivity analysis on the number of selected dynamic features.

Figure 6: A map of civil unrest events and forecasting hotspots on March 17th, 2013 in Brazil.

1, which demonstrates that even only 1 dynamic feature could be sufficient to capture the dynamic in the civil unrest tweets, and adding more dynamic features does not add extra information.

6.4 Case Studies

We observed numerous interesting events predicted by the proposed approaches, CMTFL-I and CMTFL-II in our experiments. For instance, Figures 6 and 7 record two waves of civil unrest events that occurred on March 17th, 2013 in Brazil, and April 17th, 2013 in Paraguay, respectively.

We can observe from Figure 6 that there were three events in Brazil, among which Event 1 and Event 2 happened in large cities, e.g., Sao Paulo and Rio de Janeiro, while Event 3 was in a smaller city, Niteróí. Note that the city Niteróí does not have any training sample. The proposed CMTFL-II successfully predicted all of events, even for the city Niteróí. This is because CMTFL-II jointly learned the models of all the tasks (i.e., cities). Even the model of the city has no training sample, it can still be estimated by data from other cities. The LASSO model predicted two of them but failed on the forecasting of Event 3. This is because each LASSO model is trained for each city individually, and thus the events of the city without any training sample cannot be predicted. The rMTFL model only predicted one event for Rio De Janeiro. Its failure of discovering events for two other cities might be due to its exclusion of the dynamic features after training, as shown in Table 4. This reduces its capability to uncover the burstiness of dynamic keywords. This further verifies the need for a separate selection of the static and dynamic features as in our CMTFL-II model.

We can observe from Figure 7 that there were four events in Paraguay, among which Event 2, Event 3, and Event 4 had been successfully predicted by CMTFL-II. rMTFL predicted Event 2 and Event 3 while LASSO failed to predict any event. As shown in Table 1, Paraguay is a country that the number of reported events is large but the volume of tweets is relatively small, i.e., the ratio of #tweets/#events is less than one third of other countries. The sparsity of tweets data make the forecasting more difficult for Paraguay by methods without using multi-task learning, as shown in Table 2.

7. CONCLUSIONS

This paper presents a novel multi-task learning framework to the problem of spatial event forecasting in Social Media. Existing methods are not able to concurrently address the critical challenges, such as dynamic patterns of features, and geographic heterogeneity. Our work considers the estimation of predictive models in different locations as a multi-task learning problem, in order to use the shared information between locations, which effectively increases the sample size for each location. We further model both static and dynamic features using different constraints to balance both homogeneity and diversity between these two types of features. We propose efficient algorithms based on the IGHT that are able to predict spatial events in real time. Our empirical results demonstrated that we can effectively detect civil unrest events, outperforming competing methods by a substantial margin on both precision and recall. For the future work, we plan to extend our multi-task learning framework by exploring more complex relationships between locations and integrating human domain knowledge as priors.

Acknowledgement

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via DoI/NBC contract number D12PC000337, the US Government is authorized to reproduce and distribute reprints of this work for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the US Government.

8. REFERENCES


