ABSTRACT

BAKERMAN, JORDAN. Twitter Analytics: Geotag Imputation, Forecasting, and Dynamic Variable Selection. (Under the direction of Alyson Wilson.)

The popularity of social media has created vast repositories of open source data with broad potential value. Researchers are actively mining these new complex data sources to create predictive models for wide-ranging applications. For example, Wikipedia is used to forecast influenza in the United States [Hickmann et al., 2015], Facebook is used for more effective advertising [Backstrom et al., 2010], and Twitter is used to forecast civil unrest in Latin America [Korkmaz et al., 2015]. In this dissertation, we create statistical methodology advancing the analytical value of Twitter.

We begin in Chapter 2 by developing a geotag imputation method to predict the origin of individual tweets. Standard practice uses either the content of the tweet, network information, or these two features independently to estimate the origin. We show improved accuracy by using both tweet text and user network information jointly. Moreover, we properly account for uncertainty, improving both precision and coverage of geotag imputation.

In Chapter 3 we focus on short term forecasting using daily word counts as model features scraped from Twitter. Conventional forecasting models in the area of social media are typically static, and therefore, researchers assume time invariant data. We consider a dynamic approach to account for possible time dependencies, which allows the forecasting model to evolve in time along with the data generating process. For the problem of civil unrest, we use dynamic logistic regression to forecast the probability of protest in Latin America and show improved accuracy compared to the static baseline model. Furthermore, we develop a dynamic variable selection technique based on penalized credible

regions in order to contextualize the reasons for protest. The proposed methodology is scalable and outperforms the current baseline.

In Chapter 4, we combine the geotag imputation and dynamic model methodology of the previous chapters. This final project is a first step in using tweets with imputed geotags within geographic-specific forecast models. The goal is to understand the impact of measurement error due to the location uncertainty of tweets.

Twitter Analytics: Geotag Imputation, Forecasting, and Dynamic Variable Selection

by
Jordan Bakerman

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Statistics

Raleigh, North Carolina

2018

APPROVED BY:

Eric Laber

Donald Martin

John Mattingly

Karl Pazdernik
External Member

Alyson Wilson
Chair of Advisory Committee

DEDICATION

To my father, grandparents, and wife.

BIOGRAPHY

Jordan Bakerman was born in Ludlow, Vermont. He graduated from the University of Vermont with a Bachelor of Science degree in Finance and Applied Mathematics in 2011 and a Master of Science degree in Statistics in 2012. He then joined North Carolina State University to earn a Ph.D. in Statistics.

ACKNOWLEDGEMENTS

I would like to first thank Dr. Alyson Wilson and Dr. Karl Pazdernik for their unwavering encouragement, mentoring, and patience throughout the dissertation process. I am grateful for your guidance and support. I also thank the numerous NCSU and UVM community members who have had a hand in my growth and success. In particular, thank you to Dr. Richard Single for inspiring and encouraging me to pursue a career in Statistics. Finally, I would like to thank my father and wife for their love and support.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction and Motivation

Twitter was created in 2006 as a free online social networking service. Members of the community post messages in a public forum with a maximum 140 characters called "tweets" for any users of the microblogging service to view. Users may post generic details of their personal life or more valuable information such as a hurricane or earthquake update in their local community. Due to the networking capability and ease of adoption, Twitter has grown rapidly since its inception. As of 2015, Twitter is home to more than 300 million active monthly users generating more than 500 million tweets daily [Ajao et al., 2015]. The popularity of Twitter has sparked considerable research into mining the open source data for broad potential value. In this dissertation, we contribute to the Twitter literature in four areas: geotag imputation, forecasting, dynamic variable selection, and forecasting with measurement error caused by location uncertainty of tweets.

1.1 Geotag Imputation

Each tweet can be geotagged with a latitude and longitude pair, either according to a user's cell phone geo-location service or the HTML5 geolocation API when a person tweets from their computer. However, Twitter users may remove this geotagging option, and most do so. A recent study of Twitter showed less than 1% of tweets are geotagged [Ajao et al., 2015]. Given that a majority of tweets do not contain an origin, and therefore potentially useful tweets are without geographic context, considerable research has been devoted to geotag imputation.

There are two approachs to geotag imputation in the Twitter literature. The first approach relies on mining the 1% of geotagged tweets to find spatially predictive terms. For example, the term "#WolfPackNation" is used to acknowledge North Carolina State University. To impute a geotag for a tweet that contains the term "#WolfPackNation", given no other spatially meaningful terms, it is likely the term originated near Raleigh, North Carolina. The second approach estimates location using network analytics. Researchers construct network graphs of the user posting the message and then estimate the origin of the tweet according to where users in the network typically tweet. For example, if all of ones' friends tweet from Washington, D.C., it is likely the user with an unknown origin also tweets from the nation's capital.

In the first and only hybrid geotag imputation model proposed, Rahimi et al. [2015] use the model features sequentially to estimate location. The authors first train a logistic regression model to estimate the most likely location of the tweet in a discretized map. Next, the estimates are updated using spatial label propagation, an algorithm detailed in Chapter 2, to account for the network information of the user posting the tweet. Although this method improves on the geotag imputation accuracy of the literature, it does not weight each model feature by its predictive power, nor does it account for the uncertainty in the location estimate.

In Chapter 2 we develop a hybrid geotag imputation model which uses both text and network features jointly. Our model weights the features according to their geographic scope in order to estimate location using all information simultaneously. In addition, we extend the work of Priedhorsky et al. [2014] by also using Gaussian mixture models to map the spatial distribution of each feature. This allows us to account for uncertainty by estimating the most likely origin of the tweet using a bivariate probability distribution across the earth's surface. By modeling features jointly we show a significant gain in accuracy compared to Rahimi et al. [2015] and improve on the precision and coverage metrics of Priedhorsky et al. [2014].

1.2 Forecasting and Dynamic Variable Selection

Twitter has become a powerful tool for predicting real world affairs. Researchers mine the user generated content and use daily word counts to forecast diverse outcomes. For example, terms like "flu" and "fever" are leading indicators of influenza like illness rates reported by the Centers for Disease Control and Prevention in the United States [Achrekar et al., 2012]. Similarly, daily word counts have been used to forecast stock price movement [Bollen et al., 2011], box office revenue for the film industry [Asur and Huberman, 2010], and inner city crime rates [Gerber, 2014]. The models used to forecast the aforementioned outcomes, and others, are predominantly static or use a simple autoregressive structure. For time series data, the static model does not account for correlated observations, and as a result, it ignores useful forecasting information.

In Chapter 3 we consider the problem of forecasting civil unrest in the Latin American countries Argentina, Brazil, Colombia, Mexico, Paraguay, and Venezuela. Moving away from static methodology, we adopt dynamic logistic regression to forecast the probability of future protest. Compared to the static logistic regression baseline approach, dynamic logistic regression accounts for dependencies in successive observations and therefore, is better suited for time series data. Furthermore, the intrinsic Bayesian structure of dynamic models allows parameters to be sequentially updated in time as new information becomes available. This allows forecasts to be heavily weighted towards new information and is the reason dynamic models excel in short term forecasting. We show the dynamic approach is highly advantageous as it heavily outperforms the baseline method in terms of forecast accuracy.

In order to contextualize the reasons for protest, we propose a variable selection technique for the dynamic logistic regression model. For dynamic linear models, there is considerable research on the topic of variable selection. Methodology includes latent threshold modeling motivated by the "spike-and-slab" approach [Nakajima and West, 2013] and shrinkage priors [Belmonte et al., 2014; Caron et al., 2012] motivated by the Bayesian LASSO. However, due to the challenges of fitting the model and the research dedicated thereto, there is only a single variable selection technique for dynamic generalized linear models present in the literature. This technique, Dynamic Model Averaging [McCormick et al., 2012], is only able to consider a subset of all possible candidate models similarly to its predecessor, Bayesian Model Averaging. Without knowing which terms are predictive of civil unrest, we consider nearly $1,000$ model features for a two year daily forecast period and therefore, require a scalable variable selection method. Extending the work of Bondell and Reich [2012], we use penalized credible regions to conduct variable selection dynamically. The approach is scalable and we show it improves variable selection performance compared to the LASSO regularization baseline.

1.3 Measurement Error

Geotag imputation and forecasting events with Twitter are currently separate research areas in the literature. Practitioners have yet to utilize tweets with imputed geotags in forecasting models. For some analyses, such as forecasting stock market indicators [Zhang et al., 2011], researchers are able to use tweets without an origin as location is irrelevant for the problem. For geographic specific models, such as forecasting civil unrest in Latin America [Korkmaz et al., 2015], researchers could substantially increase the supply of data by using tweets with imputed geotags. As a result, researchers may be able to improve the location granularity of the forecast event from the country level to perhaps state or city level.

In Chapter 4 we combine the geotag imputation approach of Chapter 2 and the dynamic logistic regression model of Chapter 3 in order to understand the impact of location uncertainty in forecasting with Twitter. To account for the uncertainty in imputed geotags, we build in measurement error to the covariates of the dynamic logistic regression model. Comparing dynamic logistic regression with measurement error versus assuming the imputed geotag is ground truth, we show the former enables superior forecast performance in small geographic regions.

Chapter 2

Geotag Imputation: A Hybrid Approach

2.1 Overview

Geotagging Twitter messages is an important tool for event detection and enrichment. Despite the availability of both social media content and user network information, these two features are generally utilized separately in the methodology. In this chapter, we create a hybrid method that uses Twitter content and network information jointly as model features. We use Gaussian mixture models to map the raw spatial distribution of the model features to a predicted field. This approach is scalable to large data sets and provides a natural representation of model confidence. Our method is tested against other approaches and we achieve greater prediction accuracy. The model also improves both precision and coverage.

2.2 Introduction

Twitter was created in 2006 as a free online social networking service. The service allows its users to post messages with a maximum 140 characters called "tweets" without restriction on the number of tweets sent in any given time period. As of 2015, Twitter generates more than 500 million tweets daily, representing in excess of 300 million active monthly users [Ajao et al., 2015]. The popularity of Twitter has spawned a variety of research efforts to use the micro-blogging service as a tool to support many applications, including event detection [Korkmaz et al., 2015], event monitoring [Gelernter and Mushegian, 2011], and influenza diffusion [Generous et al., 2014]. Each of these applications benefits from location information to identify and utilize relevant tweets.

Each tweet can be geotagged with a latitude and longitude pair, either according to a user's cell phone geo-location service or the HTML5 geolocation API when a person tweets from their computer. However, Twitter users may remove this geotagging option, and most users do so. A recent study of Twitter showed that less than 1% of tweets are geo-tagged [Ajao et al., 2015]. As a result, recent research has focused on estimating location information in the Twittersphere.

The possible active Twitter features are as follows. 1. **Geotagged coordinates** (latitude × longitude). 2. The **tweet** (140 characters maximum). This may include, words, symbols, toponyms, abbreviations, slang, etc. 3. **Language**. Each user chooses a language when joining Twitter, and each tweet is tagged with that chosen language. 4. **Location Field**. When joining Twitter, users may specify their location with as much or little specificity as desired. That is, the user may specify *Raleigh, NC* or *my treehouse*, both are acceptable. 5. **Time Zone**. The user may also choose their time zone. Each tweet is then tagged with a generic time zone stamp. 6. **Network**. Twitter users may "follow" other members of the community and converse with people directly using the *@username* syntax. It is through the *follower* and *followee* relationships or direct messaging that one can derive a Twitter user's network and use it as a model feature. If a particular tweet is intended for a specific user, the tweet field contains the user name. This direct engagement in an online conversation can be considered the requirement for two users to be "friends" and to create individual friendship networks.

The language, location, and time zone fields can be highly inaccurate because users may set these fields without protocol. Some authors use these fields as model features or as a test of accuracy for the ground truth and other authors only include tweets for model training with a real location field that can be mapped using a gazetteer [Cheng et al., 2010; Schulz et al., 2013; Davis Jr et al., 2011; Abrol and Khan, 2010; Rout et al., 2013]. However, an "actual" location does not mean it is the user's true location. Only 66% of users provide accurate location field information, and nearly all at the city or state level [Hecht et al., 2011].

The current state of the Twitter geotagging literature can be described according to the model features. One segment uses the tweet text only, and another uses network information only, but there is little overlap between the two methodologies. The first segment mines the training set of tweets, which are tweets with true geotagged coordinates, to find n-grams with little geographic variability. In other words, these methods identify sets of co-occuring words that are geographically narrow and predictive of location. For example, the term *Celtics* is vernacular most used in the Boston area. The network segment, on the other hand, uses the distribution of friends to predict location. This notion is predicated on the idea that Twitter relationships mimic off-line relationships [Rout et al., 2013].

Regardless of the features used for location prediction, Twitter geotagging algorithms have evolved to consider prediction accuracy as the metric for quality. Whether classifying a tweet into a predefined region or predicting the latitude × longitude origin of a tweet, current approaches can predict a tweet to within a few hundred kilometers. The literature is described more fully in Section 2.3.

In this chapter, we propose a hybrid method (Figure 2.1) that can harness the power of both text and network features. Our method proves to be highly competitive with the current state of the literature according to multiple metrics. We are able to pinpoint the origin of a tweet to within 19 km more than 50% of the time. In addition, our approach has several key advantages compared to the literature.

1. **We use a hybrid model.** We provide a method to incorporate both text and network information to obtain a more accurate location estimate. In addition, increasing the number of model features correlates to predicting a larger number of tweets. The hybrid approach can still be used if at least one of the features, text or network, is available. If a tweet does not contain any predictive text, the hybrid model will use the network information only, and vice versa.

2. **The text and network features are used jointly as predictors.** Our model weights the features according to their geographic scope. This contrasts to the only other hybrid model in the literature [Rahimi et al., 2015], which uses text and network features consecutively without regard for the predictive power of each feature.

3. **We avoid using a gazetteer.** Filtering training tweets by accurate location field information can highly reduce the training set size. We use all tweets (divided into training and test sets) tagged with a latitude × longitude pair.

4. **We avoid fabricated boundaries.** We treat geotagging as a coordinate prediction

problem and not a classification problem. We avoid creating predefined geographic regions which are subject to data sparsity difficulties.

5. **More complete preprocessing.** We take several text mining preprocessing steps to clean the data. We also remove outliers for the text features that skew predictions.

6. **We use Gaussian Mixture Models.**

   (a) **GMMs are scalable.** This modeling technique is computationally inexpensive and, as a result, scales to large data sets. It also allows for efficient testing of different component mixture models to find the best fitting model.

   (b) **GMMs are multi-modal.** Model features must be geographically narrow to represent a good predictor, but a single prediction region is not necessary. Multiple locations are estimated for a single tweet with associated probabilities to gauge each location's possibility as the origin. For example, the word *Burlington* is linked to over 20 cities in the United States.

   (c) **GMMs are interpretable probability distributions.** GMMs provide a natural way to interpret results and model error. Location estimates are bivariate probability distributions, visually displaying the most likely origin of a tweet and the model confidence in the estimate.

The chapter is organized as follows. In Section 2.3 we discuss the current state of geotagging literature and the need for a hybrid approach. Section 2.4 describes the necessary preprocessing steps to remove signal noise and Section 2.5 details the new hybrid algorithm. Section 2.6 shows the performance of the model and its utility compared to the current literature. Finally, Section 2.7 describes an example application to use geotagged Twitter data, and Section 2.8 contains conclusions.

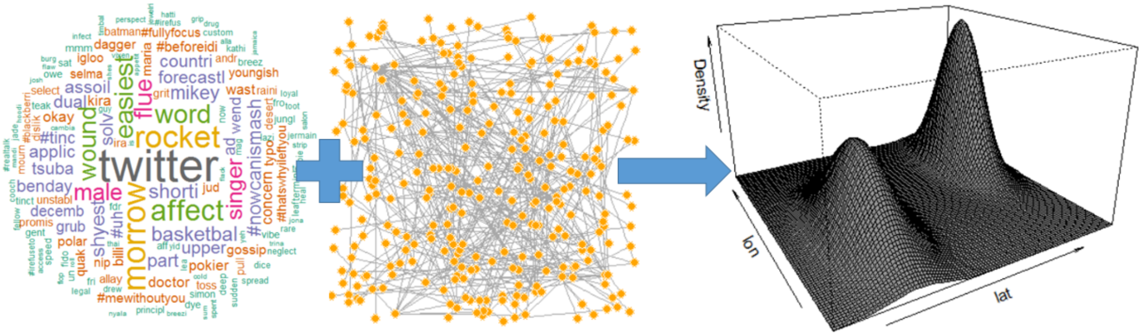


Figure 2.1: The general approach to our hybrid method. Combine Twitter text and network information to predict location with a bivariate density estimate.

2.3 Related Work

The current geotagging literature can be grouped into three categories: text, network, and hybrid methods. We briefly describe the state of the literature for each category.

2.3.1 Text Approaches

In one of the first Twitter geotagging methods, Eisenstein et al. [2010] developed a latent variable model that treats the text and location of tweets as outputs from a generative process. The authors showed that lexical variation, and text in general, can be predictive of location. Furthermore, they created a freely available data set of nearly $380,000$ tweets that we use for model comparison with the literature. The data set is described more fully in Section 2.6.

Subsequent text-based approaches regarded geotagging as a classification problem. Several authors [Cheng et al., 2010; Chandra et al., 2011; Kinsella et al., 2011; Hecht et al., 2011; Wing and Baldridge, 2011; Roller et al., 2012; Hulden et al., 2015] predict location according to a region of varying granularities, such as country, state, city, zip code, or geographic areas defined by degrees of latitude and longitude. These methods use the frequency of words within regions to train a model and predict location using Bayes Theorem. For example, naive Bayes is a simple and scalable method used to estimate the probability of a categorical event, such as the city origin of a tweet. The highest probability event is then regarded as the classification estimate.

The more recent text-based approaches move away from classification techniques to predict the latitude and longitude coordinate pair. The authors in Schulz et al. [2013] use a method called *polygon stacking* to estimate unique geographic regions. The authors use words that can be mapped to a region according to a gazetteer, which is a geographical dictionary. Words with geographical meaning, such as "Okemo Mountain," are mapped to a physical location representing a geographic polygon. Each polygon for each word in the tweet is then stacked on top of each other and the center of the tallest plateau is considered the location estimate.

In a more intuitive approach, Priedhorsky et al. [2014] use Gaussian mixture models to estimate geographic probability distributions. The authors show that the technique is scalable, multi-modal, geographically interpretable, and avoids artificial boundaries. For these reasons, the Gaussian mixture model is the basis of our new algorithm.

2.3.2 Network Approaches

There are sets of network-based methods that also considered geotagging as a classification problem. In the simplest network based approach, Davis Jr et al. [2011] classifies a tweet at the city level according to where the largest proportion of their friends are located. Abrol and Khan [2010] use the same method as Davis Jr et al. [2011], except they consider the variable depth of a user's network to classify at the city level. For example, at a variable depth of $d = 2$, the algorithm considers the locations of an individual's friends, $F_1, \ldots, F_n$, and the locations of each of $F_1, \ldots, F_n$'s friends as well. The authors in Rout et al. [2013] use support vector machines to classify at the city level in the United Kingdom using features such as city population and the number of reciprocated tweets. In addition, Rout et al. [2013] scale the population density of each city to avoid weighting the model towards the largest city, i.e., London.

As with the text-based approaches, some methods focus on classification, while others try to predict location according to the latitude and longitude coordinate pair. Although not initially intended for Twitter, Backstrom et al. [2010] establishes one of the first geotagging methodologies for social media, specifically Facebook. The authors show empirically that the probability of friendship is inversely proportional to the geographic distance between two people and use these probabilities within a likelihood approach to estimate the most probable location of a user compared to the locations of a set of other users. The authors in McGee et al. [2013] extend this method to Twitter and use regression trees to first estimate the strength of friendship between users. The strength of friendship determines which probability curve is used within the likelihood.

The most effective network-based approaches use an extension of label propagation, a common algorithm found in network analysis literature. This is a simple iterative algorithm designed to infer labels for unknown items in a network based on the community structure of the network. The unknown item receives the most frequent label from the group of items with which it shares a connection. Jurgens [2013] proposes spatial label propagation to account for the geographic distribution of a network. This method predicts unknown locations as the geometric median of the observed locations from a user's network. In addition, the spatial label propagation algorithm process is repeated in order to expand the set of predictions which can be made. Predictions in previous iterations are used as ground truth for the subsequent iterations. Compton et al. [2014] extend spatial label propagation and consider the frequency of directed tweets as edge weights.

2.3.3 Hybrid Approach

We are aware of only one hybrid geotagging approach, which uses both text and network structure. Rahimi et al. [2015] borrow techniques from the text and network approaches to form a hybrid model. First, the authors discretized the geographic space using the method detailed in Wing and Baldridge [2014]. Then, unigrams appearing in at least ten tweets are used as features to train a logistic regression model. The model is then used to predict the most likely cell for an unknown user and the center of the cell is the coordinate location estimate. Next, the authors use the spatial label propagation method proposed by Jurgens [2013]. This iteratively adjusts the estimates from the text method by using the geometric median of locations of a user's friends to predict location.

In general, a hybrid approach is advantageous because it includes additional model features. When text features are not predictive, the model can rely on network information and vice versa. A drawback of the hybrid algorithm detailed by Rahimi et al. [2015] is that the algorithm uses text and network features independently and sequentially. In doing so, it uses predictions as ground truth before beginning the spatial label propagation algorithm and does not quantify model uncertainty. Our algorithm uses both model features jointly for prediction while accounting for model uncertainty. The next two sections provide details of the algorithm and in Section 2.6 we illustrate that our method provides greater prediction accuracy when applied to a sample data set.

2.4 Preprocessing

Prior to model creation, we filter the data extensively. These steps are designed to filter noise from Twitter data and allow discovery of relationships between the model features and the geographic coordinates.

2.4.1 Text Preprocessing

The four text preprocessing steps below, in combination, reduce the number of unigrams present in the data and increase the rate of sparse words. For example, the words *cool* and *COOOOL* are considered the same unigram. Therefore, we obtain two instances of a single unigram instead of one instance of two separate unigrams. As a result, we are able to supply our subsequent analyses with more data. To illustrate the steps, consider the following tweet as an example.

- Hey @kpazphd OMG! my new car is SWEEEET!!! #TESLAlife watch out Raleigh! livin the dream...

First we remove all special characters, emojis, and punctuation from the text except for the @ symbol, which indicates a direct tweet, and the hashtag (#) symbol. In Twitter, the hashtag is used before keywords or phrases to denote a topic or theme. This effectively lets users categorize tweets and allows for simple queries to monitor trending topics in the Twittersphere. For example, *#blueandblack* and *#whiteandgold* were tweeted more than 4.4 million times over a two day period as people debated the color of a peculiar dress online. The hashtag gives the phrase an entirely different meaning and context than if we considered the phrases *blue and black* and *white and gold* as is. Simultaneously, we also reduce all characters to lower case to avoid case sensitivity. We assume case bears no predictive power.

- hey @kpazphd omg my new car is sweeeet #teslalife watch out raleigh livin the dream

Next, we remove all stop words because we assume there is no association between the most common words used in Twitter and geographic location. Stop words are simply the most common words in a given language. For example *the*, *is*, and *no* are common English stop words. However, there is no universal set of stop words. We combine two dictionaries for this preprocessing step. We combine the SMART stop word list, a list of 571 words developed by Cornell [Feinerer et al., 2008], and also the top 500 most common Twitter words uncovered by TIME magazine [TIME, 2009]. The second list is used to remove common slang terms such as *lol*, *omg*, and *wtf* from the Twitter feed.

- hey @kpazphd car sweeeet #teslalife watch raleigh livin dream

Next, we perform a spell check on the remaining words that do not start with the @ or # symbol. Thus, we assume the association between misspelled words and geographic location is the same as correctly spelled words and geographic location. Each remaining word is compared to an English dictionary for accuracy. If the word is misspelled, it is replaced with the closest matching word according to the Jaccard coefficient for strings. The Jaccard coefficient simply measures the similarity between two sets, $|A \cap B|/|A \cup B|$. In the context of language processing, A and B are two separate unigrams for which the letters are compared.

- hey @kpazphd car sweet #teslalife watch raleigh living dream

Finally, we apply the Porter stemming algorithm, a widely used English stemming algorithm [Porter, 1980]. This reduces each remaining word to its root. For example, the words *jumping*, *jumper*, and *jumped* are set to the root word *jump*.

- hey @kpazphd car sweet #teslalife watch raleigh live dream

2.4.2 Text and Network Preprocessing

The unigrams present after the preprocessing steps above are then used to create unigram variables. Each unigram variable simply represents the locations (in latitude × longitude coordinates) from which it was tweeted.

Likewise, we create each user's network by considering *direct* tweets the barometer for friendship. A directed tweet is equivalent to sending a message to a specific person in a public forum. Directed tweets are performed using the *@username* syntax to refer to a specific user. That is, *userA* sends a direct tweet to *userB* by simply including the syntax *@userB* in the tweet. In one study, Huberman et al. [2008] shows that approximately 25% of posts are directed tweets. To create *userA*'s network, we find every instance when *userA* sends a direct tweet. The users receiving the direct tweet are considered *userA*'s friends. We then search for tweets inititated by these friends that are also geotagged and include them in the network. *userA*'s network variable is now associated with a set of coordinates (latitude × longitude). This process is repeated for each user.

A final preprocessing step is applied to both the text and network variables. We cluster the data into subpopulations for outlier prepossessing using the well-known K-means clustering procedure. This algorithm clusters data into k clusters and finds their centers by minimizing the sum of Euclidean distances of all data points to their respective centers. To estimate the number of appropriate clusters for each variable, we use the Gap statistic [Tibshirani et al., 2001]. This technique chooses k by finding the largest discrepancy of the pooled within-cluster sum of squares, W_k, for the data and the expectation of W_k under a null distribution,

$$\underset{k \in \{1,\ldots,20\}}{\operatorname{argmax}} \; Gap(k) = E\{log(W_k)\} - log(W_k) \; .$$

For simplicity, we use a uniform distribution over the range of observed data for each variable.

If an observation is farther than the mean distance from the center of its subpopulation, it is considered an "outlier" and removed from the data set [Aggarwal and Singh, 2013]. For tightly clustered data this removes subpopulation anomalies. Figure 2.2 illustrates this technique.

In addition, we remove all text variables with less than ten observations to avoid modeling sparse data in the next section. This final technique is widely adopted in the geotagging literature [Eisenstein et al., 2010; Wing and Baldridge, 2011; Hulden et al., 2015; Priedhorsky et al., 2014].

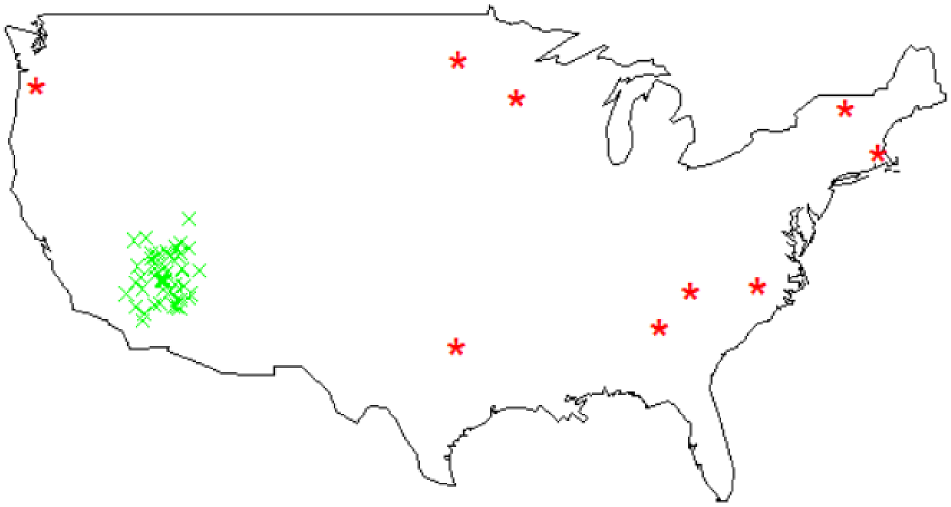


Figure 2.2: A set of text or network coordinates. The ⋆ observations are considered outliers and removed before the Gaussian Mixture Model is created.

2.5 New Hybrid Model

Our hybrid model is constructed using a series of Gaussian mixture models. We first create a bivariate density estimate for each unigram u_i and network n_j. Here, i and j represent the index for the unique unigrams and networks respectively. Then we combine the appropriate unigram and network Gaussian mixture models using an intuitive reweighting algorithm to create a final Gaussian mixture model, $h(\cdot)$. This final model is a geographic probability distribution representing an estimate of the origin location of a non-geotagged tweet. The rest of this section details the creation of $h(\cdot)$.

Let $y \in \mathbb{R}^2$ be a single pair of coordinates, latitude × longitude. For each unigram u_i we fit a Gaussian mixture model to the observed locations,

$$g(y|u_i) = \sum_{k=1}^{c_i} \phi_{ik} N(y|\mu_{ik}, \Sigma_{ik}) \ , \tag{2.1}$$

where N is the bivariate normal density function with parameters μ_{ik} and Σ_{ik} as the mean vector and covariance matrix respectively. In addition, ϕ_{ik} are the mixture weights which combine the subpopulation probability distributions into a single density estimate. The subscript k represents the individual mixture components for each GMM and c_i is the number of mixture components for unigram u_i. The parameters ϕ_{ik}, μ_{ik}, and Σ_{ik} are estimated using the expectation maximization algorithm from the *MCLUST* package in *R* [Fraley et al., 2012]. The volume, shape, and orientation of each mixture are free to vary, which implies that all three covariance matrix parameters must be estimated.

The authors in Priedhorsky et al. [2014] use a simple heuristic approach to choose the number of components for each GMM, $\min(20, \log(s)/2)$, where s is the sample size. Instead, we choose the number of components, c_i, which yields the best Bayesian Information Criterion (BIC) for the GMM when testing components $1, \ldots, 20$ individually,

$$\arg\min_{c_i} \left(-2\sum_{b=1}^{s_i} \log\left[\sum_{k=1}^{c_i} \hat{\phi}_{ik} N(y_b|\hat{\mu}_{ik}, \hat{\Sigma}_{ik}) \right] + p\log(s_i) \right) .$$

Here, p represents the number of parameters to be estimated and s_i is the sample size. We use the BIC to balance both model fit and model complexity.

Similarly, define the Gaussian mixture model for a user's network as

$$f(y|n_j) = \sum_{k=1}^{c_j} \theta_{jk} N(y|\mu_{jk}, \Sigma_{jk}) \ . \tag{2.2}$$

The number of components, c_j, as well as the parameters θ_{jk}, μ_{jk}, and Σ_{jk} are estimated in the same fashion as before. Recall from Section 2.4 that the data used to fit each network GMM are the locations where users within the network tweeted. That is, the coordinates of the network are targets to fit the GMM and estimate parameters similarly to the unigram GMMs.

Next, we combine the applicable text and network GMMs above to create a final density estimate of the origin of a tweet. Let m be the set of unique unigrams that appear in a single tweet for which individual density estimates, $g(y|u_i)$, were created. Define the hybrid model for a given tweet, m, and the network of the user initiating the tweet, n_j, as

$$h(y|m, n_j) = \delta_0 f(y|n_j) + \sum_{l=1}^{T_m} \delta_l g(y|m_l) \ , \tag{2.3}$$

where $\{\delta_0, \ldots, \delta_{T_m}\}$ represents the mixture weights. Specifically, δ_0 is the network GMM weight and $\{\delta_1, \ldots, \delta_{T_m}\}$ are the text GMM weights. In addition, T_m is the cardinality of m and m_l represents the l^{th} element of the set m. That is, m_l refers to a unique unigram of a single tweet.

A useful text and network location estimator, from Equations (2.1) and (2.2) respectively, should be geographically narrow, i.e., the subpopulations it models should be small in area. Thus, we weight the hybrid density estimate in Equation (2.3) towards the GMMs with small prediction areas while also balancing the probabilities of the mixture components. We first define the weights for the network GMM, δ_0^*, and unigram GMMs, δ_l^*, as the inverse weighted average of the mixture probabilities (θ_{jk} and ϕ_{ik}) and the area of the highest $100 \times (1-\alpha)\%$ density of each mixture component. The area of each mixture component is defined as $\pi\chi_2^2(\alpha)det(\Sigma)^{1/2}$ and is simply a function of the cumulative density, $\chi_2^2(\alpha)$, and the shape and magnitude of the ellipse, $det(\Sigma)^{1/2}$, containing $100 \times (1-\alpha)\%$ of each separate bivariate density. The initial weights δ_0^* and δ_l^* are calculated as

$$\delta_0^* = \frac{1}{\sum_{k=1}^{c_j} \theta_{jk}\pi\chi_2^2(0.05)det(\Sigma_{jk})^{1/2}} \ ,$$

$$\delta_l^* = \frac{1}{\sum_{k=1}^{c_i} \phi_{ik}\pi\chi_2^2(0.05)det(\Sigma_{ik})^{1/2}} \ . \qquad (2.4)$$

To ensure that the hybrid model is also a Gaussian mixture model, we normalize the weights, $\{\delta_0^*, \ldots, \delta_{T_m}^*\}$, for each prediction. Thus, the weights used in Equation (2.3) are

$$\{\delta_0, \ldots, \delta_{T_m}\} = \left\{ \frac{\delta_0^*}{\sum_{l=0}^{T_m} \delta_l^*}, \ldots, \frac{\delta_{T_m}^*}{\sum_{l=0}^{T_m} \delta_l^*} \right\} .$$

To illustrate the weighting algorithm, consider the location prediction of a tweet with a network and only one unigram. Let each GMM, $f(y|n_j)$ and $g(y|u_i)$, be a two component model with the mixture probabilities and associated mixture component areas $\{(\theta_{j1}, \theta_{j2}) = (0.1, 0.9), (A_{j1}, A_{j2}) = (15\,km^2, 5\,km^2)\}$ and $\{(\phi_{i1}, \phi_{i2}) = (0.5, 0.5), (A_{i1}, A_{i2}) =$

$(10\,km^2, 10\,km^2)\}$ respectively. Note, A_{jk} and A_{ik} correspond to the areas calculated as a function of Σ_{jk} and Σ_{ik}, respectively. Using Equation (2.4) to calculate the weights, $\delta_0^* = 1/6$ and $\delta_1^* = 1/10$ ($\delta_0 = 0.625$ and $\delta_1 = 0.375$ normalized). In this case, we weight the hybrid model towards the network estimate, even though the total area for each GMM is the same, because 90% of the data comes from a subpopulation with a small area, $5\,km^2$.

Figure 2.3 displays one such estimate of $h(y|m, n_j)$. In this instance, the highest probability subpopulation corresponds to the user's network, which is near Denver, Colorado. Here, the user's network is the best predictor of the origin of the tweet. The text portion of the model relates unigrams to other subpopulations throughout the country with lower probability.

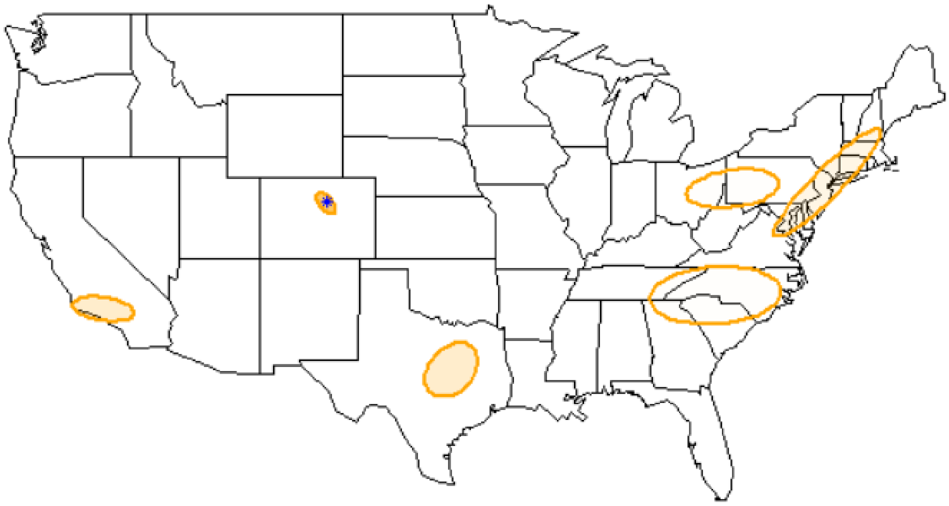


Figure 2.3: An example of the hybrid model density estimate, $h(y|m, n_j)$, predicting location of a single tweet. Each ellipse models a separate subpopulation and the darker the transparency the higher the probability. The true origin of the tweet is marked by the $\star$ symbol.

2.6 Results

In general, the hybrid model, $h(y|m, n_j)$, estimates the probability of each point on the Earth's surface, y, being the origin of the tweet, given the text contained in the tweet, m, and the user's friendship network, n_j. In this section we test our hybrid model on a common data set from the geotagging literature and evaluate the hybrid model according to several metrics. The metrics we use are described in Priedhorsky et al. [2014], who also use Gaussian mixture models for geotagging.

2.6.1 Data

The data set we use was collected by Eisenstein et al. [2010]. It was accumulated in March 2010 from Twitter's "Gardenhose" Streaming API, which was a 15% sample of all daily messages. The authors kept only geotagged data within the contiguous United States. In addition, they filter the remaining data to include users following and followed by less than 1000 people in an attempt to avoid celebrities. The final data set consists of approximately 380,000 tweets and 9,500 users. The locations of the tweets from this data set are displayed in Figure 2.4.

For subsequent analyses, we randomly split the data into 90% for training and 10% for testing. The training data is preprocessed and a model is fit as described in Sections 2.4 and 2.5. The test data is filtered through the same text preprocessing steps to maintain consistency between the two sets.

2.6.2 Performance Metrics

The *simple accuracy error* (SAE) metric measures the distance from the most probable location to the origin of the tweet. For example, the highest density estimate in Figure 2.3

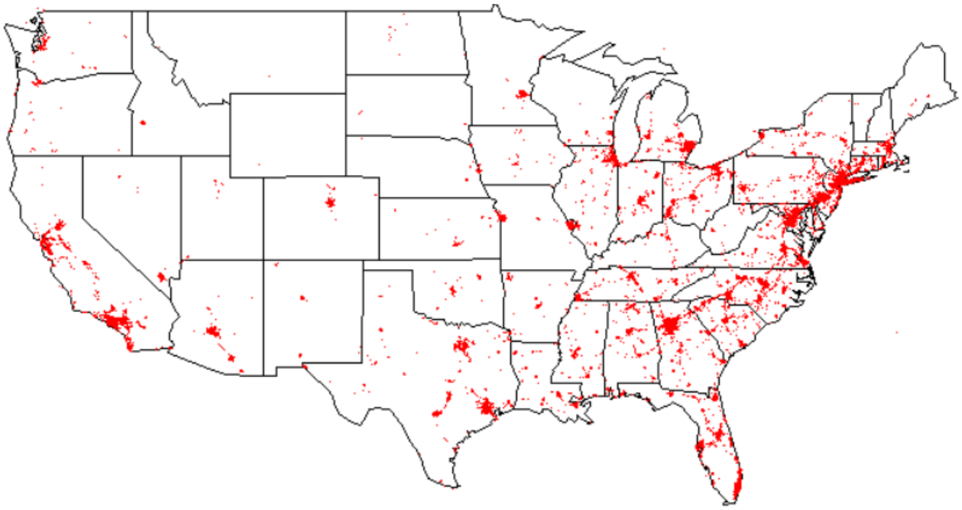


Figure 2.4: Locations of all tweets within the Eisenstein data set.

coincides with an estimate near the actual location. Letting $d(\cdot)$ be the great-circle distance function and y' be the true origin of the tweet, the SAE is defined as

$$SAE = d(\ \arg\max_{y}\ h(y|m, n_j),\ \ y'\ )\ . \tag{2.5}$$

This metric is directly comparable to prior work, most of which forego geographic probability distribution estimates. Table 2.1 contrasts the SAE for our model compared to others in the literature.

We achieve a median prediction error of only 19 km on the test data set, considerably better than most of the literature. Although the hybrid model is multi-modal by default, our algorithm maintains highly accurate single point estimates. In addition, we have a mean SAE of 593 km, which also outperforms the other algorithms in Table 2.1. The difference in the mean and median SAEs indicates that the origin of some tweets are highly difficult to predict, skewing the overall results.

Table 2.1 also shows that our algorithm outperforms the only other hybrid model

Table 2.1: The simple accuracy error metric compared to the literature using the Eisenstein data set from 2010. Results are reported in kilometers using the great-circle distance function.

Algorithm	Features	Mean SAE	Med SAE
Eisenstein et al. (2010)	Text	$845\,km$	$501\,km$
Wing and Baldridge (2011)	Text	$967\,km$	$479\,km$
Roller et al. (2012)	Text	$897\,km$	$432\,km$
Hulden et al. (2015)	Text	$765\,km$	$357\,km$
Priedhorsky et al. (2014)	Text	$923\,km$	$645\,km$
Rahimi et al. (2015)	Hybrid	$654\,km$	$151\,km$
New Hybrid Model	Hybrid	$593\,km$	$19\,km$

in the literature by a factor of nearly 8. Recall, the method presented by Rahimi et al. [2015] uses the text and network features sequentially. The authors use the text features to predict locations without a network and then perform spatial label propagation to finish the algorithm. Our model, on the other hand, uses both features jointly and weights each according to their geographic distribution. Spatial label propagation is competitive with our hybrid algorithm [Compton et al., 2014]. However, the authors heavily filter the data and only predict approximately 10% of the data for competitive results. Our algorithm is able to successfully predict nearly 99% of the test data.

To account for the geographic distribution of subpopulation mixtures, the *comprehensive accuracy error* (CAE) is a measure of the expected distance between the true origin of the tweet and a random point generated from the model. There is no requirement for subpopulations to be clustered near each other for $h(y|m, n_j)$ to be a good estimator. However, this metric indicates whether or not the best subpopulation mixture is large in probability, small in area, and near the true location. It measures how accurate the density estimate is entirely, as opposed to a single best guess. The CAE is defined as

$$CAE = E_h[d(y,y')] = \int_y d(y,y')h(y|m,n_j)dy \ . \tag{2.6}$$

To estimate the integral in Equation (2.6) we use the Monte Carlo method, $CAE \approx \frac{1}{|z|}\sum_{y \in z} d(y,y')$, where z is a sample from $h(y|m,n_j)$ of size 100 in our testing.

The *prediction region area*, denoted PRA_α, is the area encompassed by $100 \times (1-\alpha)\%$ density of $h(y|m,n_j)$. There are multiple ways to calculate PRA_α. We sum the area covered by the highest $100 \times (1-\alpha)\%$ density of each mixture contributing to $h(y|m,n_j)$. The area of each mixture is calculated as a function of α and its covariance matrix Σ,

$$\pi\chi^2_2(\alpha)det(\Sigma)^{1/2} \ .$$

A small prediction region is ideal yet unserviceable if it rarely covers the true location. Therefore the PRA_α performance is assessed concurrently with the *coverage*, COV_α, or the proportion of times the prediction region covers the true origin of the tweet. To calculate COV_α, we simply measure the proportion of times the true origin of the tweet is within the ellipses defined by PRA_α for the test set. A geographic point $y' = (y'_1, y'_2)$ is within the boundary of an ellipse with center μ and covariance matrix Σ if

$$(y' - \mu)^T\Sigma^{-1}(y' - \mu) \leqslant \chi^2_{2(\alpha)} \ .$$

The *comprehensive accuracy error*, *prediction region area*, and *coverage* metrics are unique to models with geographic distribution estimates. As a result, Table 2.2 compares these metrics with Priedhorsky et al. [2014] only.

As expected, the hybrid model heavily outperforms the text based Gaussian mixture model method. The expected distance between the true origin of the tweet and a random

Table 2.2: The comprehensive accuracy error, prediction region area, and coverage metrics compared to Priedhorsky et al. [2014] because both algorithms use Guassian mixture models. The weighting scheme applied by Priedhorsky et al. [2014] is the sum of the product of the elements in each covariance matrix of each GMM.

Metric	Priedhorsky et al. (2014)	New Hybrid
Mean CAE	$1,445\,km$	$557\,km$
Median CAE	$1,205\,km$	$117\,km$
Mean $PRA_{0.95}$	$3,156,517\,km^2$	$251,120\,km^2$
Median $PRA_{0.95}$	$2,846,610\,km^2$	$183,954\,km^2$
$COV_{0.95}$	0.99	0.94

point generated from our model is less than 117 km, 50% of the time. Similar to the discrepancy between the mean and median SAE, the difference between the mean and median CAE indicate the predictions are skewed. However, this simply means that some subpopulations in $h(\cdot)$ are a large distance from one another. For example, if a person moves from Atlanta, Georgia to Las Vegas, Nevada, it is likely the hybrid model will estimate a subpopulation for each location, causing the CAE to be large by default. More importantly, the bimodal structure of the model captures both possible locations.

The mean and median $PRA_{0.95}$ are approximately $200,000\,km^2$. Although this is a large geographic area, it is more than ten times smaller than the $PRA_{0.95}$ of Priedhorsky et al. [2014]. To make it tangible, consider that the median prediction region of Priedhorsky et al. [2014] is more than a third of the contiguous United States. The median prediction region for the hybrid model, on the other hand, is approximately the geographic area of only North and South Carolina combined. Additionally, the coverage for the hybrid model is nearer to the nominal level in our experiments, whereas the previous method has much higher than nominal coverage due to extremely large prediction regions.

Achieving nominal coverage for the hybrid model was challenging. Initial experiments and methodology underestimated this metric by 3 – 14%. We eventually chose to sum the area covered by the highest $100 \times (1 - \alpha)\%$ density of each mixture contributing to the hybrid model. This technique achieves a coverage nearly equal to the nominal rate and also keeps the *prediction region area* metric small.

Coverage is also closely associated with the outlier preprocessing described in Section 2.4. Failure to remove outliers widens the Gaussian mixture model's prediction regions causing the coverage to inflate. We also examined the effectiveness of removing observations

- farther than 200 miles from at least 5 other observations,
- within a cluster of less than five observations where the number of subpopulations is determined by the "elbow" phenomenon in the K-means clustering algorithm.

We ultimately chose to remove observations from both text and network variables farther than the mean distance from their respective subpopulation center. Although this technique provides similar results to the aforementioned methods, it is an automated routine for removing outliers.

2.7 Application

One of the more common applications of Twitter data is the monitoring and prediction of influenza and disease outbreaks around the world. In the United States, practitioners use daily geotagged tweets containing influenza relevant keywords, such as flu, fever, cough, etc., to monitor the health of the country [Lee et al., 2013]. The authors in Achrekar et al. [2011] show daily word counts relevant to influenza can be used as a leading indicator to predict the Center for Disease Control and Prevention's (CDC) influenza-like-illness

(ILI) reports. That is, the frequency of influenza keywords are used as a proxy for the true rate of influenza among US citizens.

In Brazil, researchers are using Twitter to monitor the rate and study the diffusion of Dengue fever, a mosquito-borne illness that can lead to death if untreated. In countries without efficient government agencies to monitor the spread of disease, it is important for researchers to develop tools that can identify regions of disease outbreak in order to allocate resources properly. Gomide et al. [2011] use geotagged tweets relevant to Dengue fever to cluster data into regions. The frequency of tweets is used as a barometer for the severity of Dengue fever within each region. In an effort to utilize more information, Davis Jr et al. [2011] classify non-geotagged tweets at the city level using network information. This allows the authors to use Dengue fever relevant tweets that are not geotagged in their analyses.

In both applications, influenza in the United States and Dengue fever in Brazil, the hybrid method is advantageous. First, the hybrid method allows practitioners to geotag more data because both the text and network features are available to be used as possible predictors. In our experiments in Section 2.6, the hybrid method was able to geotag 98.2% of test tweets. Using text or network features independently, only 79.7% and 90.1% of test tweets were able to be geotagged respectively. Furthermore, the use of model uncertainty provides more information regarding the location of disease outbreak. That is, the smaller the prediction regions of the hybrid method, the more confident we are in the estimated location. Additionally, the greater the certainty in the location of a set of geotagged tweets, the greater the evidence for allocating resources to specific regions.

As an example, consider monitoring influenza in the United States using the Eisenstein data set from Section 2.6. Recall, this data set was accumulated in March 2010 from the Twitter API and only geotagged tweets in the contiguous 48 states were considered. From the $376,510$ tweets, only $1,031$ contain influenza related keywords. As evidenced by the performance metrics, the hybrid model can be used to accurately predict the location of influenza related tweets. However, the uncertainty associated with the estimates is ignored if the best predictions are simply binned into regions.

To account for the uncertainty in the predictions, suppose that we stack the geographic density estimates, $h(\cdot)$, for each predicted tweet and reweight to obtain a final geographic probability distribution of the prevalence of influenza throughout the country. Let $h_q(\cdot)$ be the q^{th} hybrid model density estimate of a sample of non-geotagged tweets S. The final distribution which combines all predictions and uses the uncertainty of each is $\sum_{q=1}^{|S|} h_q(\cdot)/|S|$. Figure 2.5 displays the estimated distribution of influenza in the United States during March of 2010 according to the Eisenstein data set.

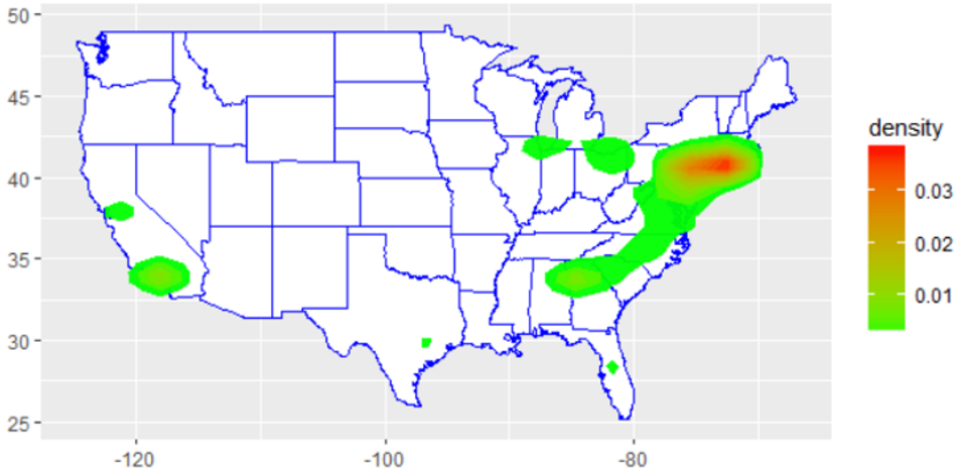


Figure 2.5: Geographic probability distribution of influenza in the United States during March of 2010.

To compare the estimated distribution of influenza to ground truth, we use the weekly CDC influenza report for the week of February 28th to March 6th, 2010 [CDC, 2010], the week the Eisenstein data was collected. In general, these reports summarize by region the percent of medical patients with influenza like illness, percent testing positive for influenza, and the number of jurisdictions reporting influenza activity. For the first week of March, only region 1 (CT, ME, MA, NH, RI, VT) and region 4 (AL, FL, GA, KY, MS, NC, SC, TN) reported widespread influenza activity. In addition, the Midwest and Southwest experienced sporadic influenza activity, and the Northwest reported no influenza activity. The report is similar to the estimated distribution in Figure 2.5. That is, there is elevated influenza in the northeast and southeast regions. In addition, there is activity in the Midwest and no activity in the Northwest. However, Figure 2.5 indicates possible elevated influenza activity in southern California; the CDC report does not support this finding.

Note, the results here do not simply imitate the most dense locations of Twitter users from the Eisenstein data set (Figure 2.4). Also, Twitter usage has grown significantly since the data was collected in 2010, but there is no reason to suspect the application is no longer suitable.

2.8 Conclusion

In this chapter we presented a hybrid geolocation model for Twitter. Our model exploits both text and network features and weights the features according to their geographic scope. Our method outperforms other geotagging algorithms according to four metrics. In particular, the median distance between the most probable location to the origin of a single tweet is only 19 km on average in our experiments.

Additionally, our hybrid model is one of only two geotagging methods which quantifies uncertainty using Gaussian mixture models. It estimates the probability of each point within a spatial domain of being the origin of a tweet. This structure allows us to visually interpret model confidence for a single tweet (Figure 2.3) or combine the uncertainty for a set of geotagged tweets to monitor an event such as influenza (Figure 2.5).

Any analysis of Twitter data is subject to the biases and limitations of the data itself. The limitations of Twitter data in general are attributable to the demographics of its users, Twitter's streaming API sampling scheme, and users turning off their location services when tweeting. First, a 2012 investigation by the Pew Research Center found that age is inversely related to the likelihood of using Twitter [Duggan and Brenner, 2013]. Internet users in the age group 18-29 are the most likely to use Twitter, and users 65 or older are the least likely. Women are more likely than men and urban residents are more likely than suburban or rural dwellers to use Twitter. Also, different levels of education and levels of household income correspond to similar rates of Twitter use. As a result of the Pew study, we know any data gathered from Twitter does not mimic the population in general.

To obtain a free sample from Twitter any person can connect to the streaming API and download approximately 1% of all tweets daily. Procuring additional tweets is a substantial cost and therefore, most researchers opt for the free sample. However, the algorithm used to sample from the streaming API is currently unknown and may not be uniformly random. Morstatter et al. [2013] show that tweets collected freely are generally biased according to trending topics and the most used hashtag strings. One final additional point of concern is that to our knowledge there is no study describing the user demographics for those who are more willing to leave the Twitter location services turned on. There may be revealing information simply from tweets with or without geotags.

The aforementioned limitations result in the scenario where the probability of detecting anomalous events on Twitter and being able to accurately impute geotags for those tweets is very low. However, Twitter data is a good source of information to monitor national or regional events. For example, an event such as the spread of influenza, as discussed in Section 2.7, is assumed to affect the population demographics equally and affects a significant proportion of Twitter users which will overcome the bias of the Streaming API. Also, people with the flu are assumed no more or less likely to turn off their location services. Events like the spread of flu overcome the limitations of Twitter data.

Chapter 3

Dynamic Logistic Regression Variable Selection

3.1 Overview

In this chapter we consider the problem of variable selection for the dynamic logistic regression model. We propose using penalized credible regions to select parameters of the updated state vector. This method avoids the need for shrinkage priors, is scalable to high-dimensional dynamic data, and allows the importance of variables to vary in time as new information becomes available. To fit the dynamic logistic regression model we utilize the Pólya-Gamma latent variable approach to improve computation time for posterior simulation. This technique alters the joint posterior distribution, providing pseudo-Gaussian data which allows the utilization of the Forward Filtering Backward Sampling algorithm, a Kalman filter based technique, to sequentially update states. We apply the proposed model fitting and variable selection methodology to the problem of civil unrest in Latin America using daily protest related terms scraped from Twitter as model features. We show improved accuracy of forecasting protests compared to the current baseline approach and report the most common predictive terms to contextualize the reasons for civil unrest. The accuracy of the variable selection technique used for the application is demonstrated by means of simulation.

3.2 Introduction

Dynamic linear models (DLMs) are utilized for forecasting complex non-stationary time series. Originally developed within engineering in the 1960s [Petris et al., 2009], a variety of applications can now be found in, for example, ecology [Calder et al., 2003], medicine [West et al., 1999], and finance [Koop and Korobilis, 2012]. The time-varying parameters of these state space regression models allow for greater flexibility in short-term forecasting. The time-varying structure allows for parameters to evolve along with structural changes in a system over time. In addition, the intrinsic Bayesian framework of DLMs allows for sequential and efficient updating of model parameters as new information becomes available.

Consider the univariate DLM specified by the *observation equation* (Equation 3.1a) and *state equation* (Equation 3.1b) for $t \geqslant 1$,

$$Y_t = \boldsymbol{X}_t \boldsymbol{\beta}_t + v_t \qquad v_t \sim N(0, \sigma^2) \ , \tag{3.1a}$$

$$\boldsymbol{\beta}_t = \boldsymbol{G}_t \boldsymbol{\beta}_{t-1} + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim N(\boldsymbol{0}, \boldsymbol{W}) \ . \tag{3.1b}$$

Y_t is the observed scalar at time t, $\boldsymbol{\beta}_t$ is the p-dimensional parameter vector (also called the state vector), $\boldsymbol{X}_t$ is a vector of known covariates, and $\boldsymbol{G}_t$ is a known $p \times p$ transition matrix governing the system disturbances, or the changes in the true underlying model. In addition, v_t and $\boldsymbol{w}_t$ are two independent sequences of independent Gaussian errors with mean zero and known variance components.

Fitting dynamic linear models has become somewhat straightforward with modern computing power and MCMC techniques, so more recent work focuses on effect selection, or determining which elements of $\boldsymbol{\beta}_t$ are nonzero at each successive time point t. With applications in finance, such as equity premium and inflation forecasting [Kalli and Griffin, 2014], it is common to have a high-dimensional state vector where many elements are unrelated to the target. Erroneous effects may reduce prediction accuracy and hinder model inference. This has spurred efforts to intelligently remove irrelevant predictors from hypothesized models. Because DLMs have advantageous properties in the Bayesian framework, such as sequentially updating the model as new information becomes available, many of the variable selection methods rely on sparsity-promoting priors. These are priors that shrink parameters towards zero if evidence suggests the predictors are unassociated with the target [Park and Casella, 2008]. Variable selection techniques will be discussed more fully in Sections 3.3 and 3.4.

Just as linear models were extended to generalized linear models by Nelder and Baker [1972], dynamic linear models were extended to dynamic generalized linear models (DGLMs) for time series data by West et al. [1985]. The DGLM allows modeling of non-Gaussian time series data such as exponential, Weibull, or gamma distributed data and discrete distributions, such as binomial and Poisson count data, all the while maintaining dynamic model structure. DGLMs are simply an extension of GLMs where model parameters are allowed to evolve over time. Similarly to the DLM, this means that these models are well-suited for short term forecasting, as parameters evolve along with structural changes in a system. Applications of the DGLM can be found in environmental statistics to monitor pollutant exposure [Chiogna et al., 2002], medicine to monitor surgical outcomes [McCormick et al., 2012], and engineering to monitor the output of a machine [Raftery et al., 2010].

Although a powerful and flexible modeling tool for non-Gaussian time series data, DGLMs have received far less attention in the literature than their predecessor, the DLM. One possible reason is that exponential family distributions are more challenging to model in the dynamic scenario. Moving away from Gaussian distributions causes online estimation of states, or sequential updating of model parameters, to become more difficult. In fact, most of the research pertaining to DGLMs focuses on simply fitting the model. Techniques include MCMC based approaches [Gamerman, 1998], data augmentation [Windle et al., 2013], and online estimation of states [West et al., 1985], which inevitably requires some degree of approximation to maintain real-time estimation.

As a result of the challenges posed from simply fitting the DGLM, little attention has been given to DGLM variable selection. One of the only DGLM variable selection methods uses a variation of Bayesian model averaging techniques [Raftery et al., 2010], which does not scale efficiently as the number of predictors grows. There is simply not a rich set of variable selection methods for the DGLM as there are with static linear models, static generalized linear models, or even dynamic linear models. In this chapter, we present a variable selection method for the dynamic logistic regression model. The *observation equation* for this DGLM can be specified as

$$y_t|\boldsymbol{\beta}_t \sim Bernoulli(\pi_t), \quad \pi_t = \frac{e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}}{1+e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}} \quad t = 1, \ldots, T \ , \tag{3.2}$$

where the response $y_t \in \{1, 0\}$ represents a “success” or “failure” respectively at time t, $\boldsymbol{\beta}_t$ is the p-dimensional state vector, and $\boldsymbol{X}_t$ is the vector of known covariates. The *state* *equation* is the same as in the DLM scenario in Equation 3.1b.

Although the motivation for this problem exists in both literature and application, as described above, our specific interest comes from the article “Combining Heterogeneous Data Sources for Civil Unrest Forecasting” [Korkmaz et al., 2015]. In this work, the authors seek to forecast the probability of protest events in six South American countries using open sources of information as model features including social media, blogs, news, and currency exchange rates. The authors use logistic regression coupled with LASSO regularization to forecast the probability of civil unrest and find Twitter key words are commonly selected as model predictors. We seek to improve upon the forecast performance by accounting for dependencies in successive observations and the possibility of changing model features over time. We also seek to find a sparse representation of the model as the application is within the $p > n$ scenario and to better infer the reasons for civil unrest.

This chapter is organized as follows. In Section 3.3 we discuss variable selection methods for static models and show in Section 3.4 how these techniques are extended to dynamic models. In Section 3.5 we discuss fitting the dynamic linear model using the Forward Filtering Backward Sampling algorithm and then segue briefly into methods for fitting dynamic generalized linear models. In Section 3.6 we propose a dynamic logistic regression model and variable selection technique and review its performance alongside the logistic regression with LASSO regularization model in Section 3.7. Finally, we apply the model to the application of forecasting civil unrest in Latin America in Section 3.8 and end with a discussion in Section 3.9.

3.3 Static Variable Selection Methods

We briefly describe some of the more popular techniques for variable selection in both the Bayesian and non-Bayesian literature. Some of the more recent methods for static models (both linear and generalized linear) are the basis for variable selection in the dynamic linear model.

First, consider the traditional linear model,

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \, , \tag{3.3}$$

where $\boldsymbol{Y}$ is the n-dimensional observation vector, $\boldsymbol{X}$ is the $n \times p$ design matrix, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ is the p-dimensional vector of model parameters, and error $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Sigma} = \sigma^2 I_n)$.

For linear regression, information criteria such as AIC [Akaike, 1974] and BIC [Schwarz, 1978] are used to compare a set of hypothesized models. These metrics assess model fit according to the maximum value of the likelihood function and are penalized by the number of estimated predictors,

$$AIC = 2k - 2\log\left(\hat{L}(\boldsymbol{\beta})\right) \qquad \text{and} \qquad BIC = \log(n)k - 2\log\left(\hat{L}(\boldsymbol{\beta})\right) \, .$$

Here, k is the number of estimated parameters, n is the number of observations, and $\hat{L}(\cdot)$ is the evaluated likelihood. The penalties discourage overfitting because goodness of fit inevitably improves as the number of predictors increases. The AIC and BIC are measures of the tradeoff between goodness of fit and model dimension. The preferred model from the candidate set is the one with the lowest information criteria.

In a more methodical approach to model selection, the AIC and BIC are used in forward and backward selection. In forward selection, one begins with the null model and iteratively adds parameters until the information criteria is no longer reduced. Backward selection is the opposite. It begins with all parameters in the model and iteratively removes parameters until the information criteria is no longer reduced.

Forward and backward selection can be combined into stepwise selection, where parameters can be added or removed at each step. The advantage of stepwise selection is that it is an automated process of choosing model predictors. However, stepwise selection does not always improve prediction accuracy. To overcome this pitfall, Tibshirani [1996] introduced the least absolute shrinkage and selection operator (LASSO). This technique involves altering the model fitting process itself by penalizing the likelihood function. $\boldsymbol{\beta}$ is chosen to minimize

$$L(\boldsymbol{\beta}) = \| \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} \|_2^2 + \lambda P(\boldsymbol{\beta}) \ ,$$

where λ is a threshold chosen by cross-validation and $P(\boldsymbol{\beta}) = \| \boldsymbol{\beta} \|_1$ is the regularization penalty. The constraint imposed on the likelihood function is used to avoid over fitting the model. The constraint forces some parameters to zero, effectively resulting in a more interpretable model. Due to the success of the LASSO method, a variety of other penalized likelihood functions have been proposed by altering $P(\boldsymbol{\beta})$, including group LASSO [Yuan and Lin, 2006], adaptive LASSO [Zou, 2006], fused LASSO [Tibshirani et al., 2005], elastic net [Zou and Hastie, 2005], smoothly clipped absolutely shrinkage and selection operator [Fan and Li, 2001], and the Dantzig selector [Candes and Tao, 2007]. Each penalty provides its own unique advantage regarding regularization.

These variable selection techniques for linear regression have also been applied to generalized linear models. Model comparison, stepwise selection, and regularization approaches are all readily exploitable in existing software. For example, the R package `glmnet` fits generalized linear models with the LASSO penalty, $\|\boldsymbol{\beta}\|_1$, the ridge penalty, $\|\boldsymbol{\beta}\|_2^2$, and a linear combination of the two known as elastic net, $(1-\lambda)\|\boldsymbol{\beta}\|_1 + \lambda\|\boldsymbol{\beta}\|_2^2$.

In parallel to non-Bayesian approaches, Bayesian variable selection for both linear regression and generalized linear models has proceeded similarly. Motivated by classical hypothesis testing, Bayes factors are used as a model comparison and selection tool. For example, consider comparing the plausibility of two models, M_1 and M_2. Kass and Raftery [1995] suggest computing the posterior odds of the two models,

$$\frac{\pi(M_1|\boldsymbol{Y})}{\pi(M_2|\boldsymbol{Y})} = \frac{\pi(\boldsymbol{Y}|M_1)}{\pi(\boldsymbol{Y}|M_2)}\frac{\pi(M_1)}{\pi(M_2)} \ ,$$

where the equation above can be interpreted as `posterior odds` = `Bayes factors` × `prior odds`. A large value of the posterior odds suggests support for the model M_1 compared to model M_2.

Considering a set of K models, $(M_1, \ldots, M_K)$, it is commonplace to choose and deploy the model with the best posterior odds after a series of pairwise comparisons. However, choosing a single best model may be inadequate. Suppose the other $K-1$ models also provide a reasonably good fit to the data as well. Hoeting et al. [1999] suggest Bayesian model averaging (BMA) to incorporate information from each plausible model. That is, simply weight the prediction from each model, $\hat{\boldsymbol{Y}}_{M_k}$, by the posterior model probability,

$$E[\hat{\boldsymbol{Y}}|\boldsymbol{y}] = \sum_{k=1}^{K} \hat{\boldsymbol{Y}}_{M_k} \pi(M_k|\boldsymbol{Y}) \ .$$

Averaging predictions in BMA allows practitioners to better account for model uncertainty [Hoeting et al., 1999].

In practice, considering all possible 2^p candidate models and computing posteriors probabilities for each is infeasible. Methods such as Gibbs variable selection (GVS) [Dellaportas et al., 2002] and stochastic search variable selection (SSVS) [George and McCulloch, 1993] have been proposed to avoid enumerating all possible models. These approaches introduce latent variables, $\boldsymbol{I} = (I_1, \ldots, I_p)$, into the model where $I_j = 1$ indicates predictor j is included in the model. Conditional on the latent variables, mixture prior distributions are placed on the model parameters, $\pi(\boldsymbol{\beta}|\boldsymbol{I})$. More commonly, the prior is referred to as "spike-and-slab". The "spike" refers to the probability $I_j \neq 0$ and the "slab" refers to the prior distribution attributed to β_j. Using MCMC techniques, these methods sample through the model space identifying posterior inclusion probabilities of each parameter, which can then be used for variable selection.

More recent literature has focused on adaptive shrinkage priors to achieve effect selection. These prior distributions shrink elements of $\boldsymbol{\beta}$ towards zero if the data support the claim $\beta_j = 0$ and avoid shrinkage for data supported evidence of $\beta_j \neq 0$. The priors are adaptive in the sense that regularization adheres to data-driven evidence and the degree of regularization can be tuned by altering prior distribution parameters, a concept similar to altering the threshold parameter λ in non-Bayesian regularization. For example, Park and Casella [2008] propose the Laplace prior which assumes a hierarchical prior for β_j,

$$\beta_j|\tau_j \sim N(0, \tau_j) \qquad \tau_j \sim exp(\delta) \ .$$

In this case, the degree of sparseness, or simply the number of $\beta_j = 0$, is tuned by the hyper-parameter δ. This method is referred to as the Bayesian LASSO.

To obtain even greater control over the degree of sparseness, Griffin et al. [2010] suggest using the Normal-Gamma prior. Again, conditional on τ_j, $\beta_j \sim N(0, \tau_j)$ and now τ_j is Gamma distributed, $\tau_j \sim Gamma(\lambda, \delta)$. The inclusion of the second parameter, λ, gives greater control over the shrinkage of model parameters. Specifically, as λ decreases, more prior mass is placed close to zero while simultaneously maintaining heavy tails in the distribution. This allows for greater shrinkage and also, allows estimated coefficients to vary in magnitude with less restriction.

3.4 DLM and DGLM Variable Selection Methods

Recall that the univariate DLM is specified by the observation equation and state equation as given in Equations 3.1a and 3.1b.

From a non-Bayesian perspective, the only dynamic model variable selection methods reside in the compressed sensing (CS) literature. In general, CS refers to reconstructing a signal in underdetermined linear systems, or simply the $p > n$ paradigm. Static CS models use variable selection techniques such as LASSO. Angelosante et al. [2009] consider dynamic compressed sensing, with time varying parameters, and propose the dynamic LASSO,

$$\underset{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_t}{\operatorname{argmin}} \sum_{t=1}^{T} \|y_t - \boldsymbol{X}_t \boldsymbol{\beta}_t\|_2^2 \qquad \text{subject to } \sum_{t=1}^{T} \|\boldsymbol{\beta}_t\|_1 \leqslant \lambda \; ,$$

where λ is the tuning parameter controlling the degree of sparseness.

Apart from compressed sensing, the rest of the dynamic model selection literature is strictly Bayesian. The Bayesian framework allows for efficient sequential updating of the state vector as new information becomes available. In fact, compressed sensing practitioners have even started to apply Bayesian methodology citing the ease of implementation [Sejdinovic et al., 2010].

Extending the concept of BMA to the dynamic setting, Raftery et al. [2010] propose dynamic model averaging (DMA) and Koop and Korobilis [2012] apply the methodology to forecast inflation. The key difference between BMA and DMA is that DMA allows the true model to vary in time. The model observation equation and state equation are specified as

$$y_t = \boldsymbol{X}_t^{(k)} \boldsymbol{\beta}_t^{(k)} + v_t^{(k)} \qquad v_t^{(k)} \sim N(0, \sigma^{2(k)}) ,$$

$$\boldsymbol{\beta}_t^{(k)} = \boldsymbol{G}_t^{(k)} \boldsymbol{\beta}_{t-1}^{(k)} + \boldsymbol{w}_t^{(k)} \qquad \boldsymbol{w}_t^{(k)} \sim N(\boldsymbol{0}, \boldsymbol{W}^{(k)}) ,$$

where k denotes the model index. DMA requires calculating the probability of each model being the true model at time t and averaging forecasts using posterior model probabilities,

$$E[\hat{y}_t | \boldsymbol{y}_{1:t-1}] = \sum_{k=1}^{K} \hat{y_t}^{(k)} \pi(M_k | \boldsymbol{y}_{1:t-1}) .$$

As before, considering all possible 2^p models is not feasible, and as a result, both BMA and DMA consider only a small set of candidate models. More recent and efficient Bayesian approaches rely on fitting the model with all parameters and shrinking elements of $\boldsymbol{\beta}_t$ toward zero in accordance with the data. Motivated by the “spike-and-slab” approach, Nakajima and West [2013] propose shrinking parameters to zero if their absolute value falls below a threshold at any point in time t. This latent threshold modeling (LTM) approach introduces a matrix of latent variables, $\boldsymbol{I}_t = diag(I_{1t}, \ldots, I_{pt})$, into the observation equation,

$$y_t = \boldsymbol{X}_t (\boldsymbol{I}_t \boldsymbol{\beta}_t) + v_t \qquad v_t \sim N(0, \sigma^2) ,$$

where $I_{jt} = I(|\beta_{jt}| \geqslant d_j)$ and $d_j \geqslant 0$ for all p. The degree of sparseness is controlled by tuning the elements of $\boldsymbol{d} = (d_1, \ldots, d_p)$.

A few adaptive shrinkage priors have been discussed in the DLM literature. Motivated by the shrinkage methodology of Park and Casella [2008], Belmonte et al. [2014] extend the Bayesian LASSO to the DLM. The authors first break the observation equation into both static parameters, $\boldsymbol{\beta}$, and dynamic parameters, $\boldsymbol{\beta}_t$,

$$y_t = \boldsymbol{X}_t\boldsymbol{\beta} + \boldsymbol{X}_t\boldsymbol{\beta}_t + v_t \qquad v_t \sim N(0, \sigma^2) \ .$$

Then, Bayesian LASSO shrinkage is applied to both the static coefficients, $\boldsymbol{\beta}$, and the variance components of the state equation, $\boldsymbol{W} = diag(\sigma_1^2, \dots, \sigma_p^2)$. That is,

$$\beta_j|\tau_j \sim N(0, \tau_j) \qquad \tau_j \sim exp(\delta_j) \ ,$$

$$\sigma_j^2|\xi_j \sim N(0, \xi_j) \qquad \xi_j \sim exp(\gamma_j) \ .$$

Due to the altered state equation and the shrinkage priors, there are three possible post model fitting scenarios.

- If σ_j^2 and β_j are shrunk to zero, then predictor j is removed from the model.
- If σ_j^2 is shrunk to zero but β_j is not shrunk to zero, then parameter β_j is static.
- If σ_j^2 is not shrunk to zero, then parameter β_j is considered dynamic.

Thus, the authors not only seek a parsimonious model but also an understanding of which parameters are static versus dynamic.

Other adaptive shrinkage priors in the DLM literature are applied only to the predictor coefficients. For example, Caron et al. [2012] place the following multivariate hierarchical prior on $\boldsymbol{\beta}_t$,

$$\boldsymbol{\beta}_t|\tau \sim N(\boldsymbol{\mu}, \tau\boldsymbol{\Sigma}) \qquad \tau \sim GiGauss(\nu, \delta, \gamma) \ ,$$

where $\boldsymbol{\mu} \in \mathbb{R}^p$, $\boldsymbol{\Sigma}$ is a $p \times p$ covariance matrix, and $GiGauss(\cdot)$ is the generalized inverse Gaussian distribution [Barndorff-Nielsen and Shephard, 2001]. Letting $\boldsymbol{\beta}_j = (\beta_{j1} \ldots, \beta_{jT})$ denote the evolution of the j^{th} parameter, then $\boldsymbol{\beta}_j \in \mathbb{R}^T$ follows the multivariate generalized hyperbolic distribution, which simplifies to static model adaptive shrinkage priors under specific parameterizations. For example, if $\delta = 0$ and $\nu = 1$ the prior reduces to the Laplace prior of Park and Casella [2008]. Likewise, if $\delta = 0$, $\nu \neq 1$, and $\nu > 0$ the prior reduces to the Normal-Gamma prior of Griffin et al. [2010].

The final adaptive shrinkage prior in the DLM literature is referred to as the Normal-Gamma Autoregressive (NGAR) process [Kalli and Griffin, 2014]. As its name implies, this prior is motivated by the Normal-Gamma prior of Griffin et al. [2010] with an extension to the DLM. Again, for this process denote $\boldsymbol{\beta}_j = (\beta_{j1} \ldots, \beta_{jT})$ as the evolution of the j^{th} predictor coefficient. The process is written $\boldsymbol{\beta}_j \sim NGAR(\lambda_j, \mu_j, \varphi_j, \rho_j)$, with the NGAR process for $\boldsymbol{\beta}_j$ defined as

$$\begin{aligned}
\kappa_{j(t-1)}|\psi_{j(t-1)} &\sim Poisson\left(\frac{\rho_j(\lambda_j/\mu_j)\psi_{j(t-1)}}{1-\rho_j}\right) , \\
\psi_{jt}|\kappa_{j(t-1)} &\sim Gamma\left(\lambda_j + \kappa_{j(t-1)}, \frac{\lambda_j}{\mu_j(1-\rho_j)}\right) , \\
\eta_{jt}|\psi_{jt} &\sim N\big(0, (1-\varphi_j^2)\psi_{jt}\big) , \\
\beta_{jt} &= \sqrt{\frac{\psi_{jt}}{\psi_{j(t-1)}}}\varphi_j\beta_{j(t-1)} + \eta_{jt}, \qquad\qquad t = 1, \ldots, T \ .
\end{aligned}$$

The process begins with $\psi_{j0} \sim Gamma(\lambda_j, \lambda_j/\mu_j)$ and $\beta_{j0}|\psi_{j0} \sim N(0, \psi_{j0})$.

The authors show, similar to the static model with a Normal-Gamma prior, the parameter λ_j controls the degree of sparseness. Small values of λ_j place more prior mass at zero and cause heavier shrinkage for the j^{th} coefficient. The autocorrelation parameters ρ_j and φ_j control the dependence between $\psi_{j(t-1)}$ and ψ_{jt}, as well as $\beta_{j(t-1)}$ and β_{jt} respectively. Thus, ρ_j and φ_j control the ability for the importance of parameters to vary in time.

There is only one dynamic generalized linear model variable selection method present in the literature. McCormick et al. [2012] extend the dynamic model averaging approach of Raftery et al. [2010] to the dynamic logistic regression model. Given a set of K candidate models, $(M_1, \ldots, M_K)$, and letting L_t be the model indicator at time t, the observation equation becomes

$$y_t | L_t = M_k \sim Bernoulli(p_t^{(k)}), \qquad \text{and} \qquad \text{logit}(p_t^{(k)}) = \boldsymbol{X}_t^{(k)} \boldsymbol{\beta}_t^{(k)} \ .$$

The forecasts from each of the K models are then averaged using $\pi(L_t = M_k | \boldsymbol{y}_{1:t-1})$ as weights.

3.5 Model Fitting

Fitting a dynamic linear model is relatively straightforward. The Bayesian paradigm allows for sequential updating of states via the Kalman filter. Fitting more complex models relies on an extension of the Kalman filter and an MCMC Gibbs algorithm framework. We first discuss fitting the DLM and then show how the methodology can be extended to fit a DGLM.

3.5.1 Fitting the Dynamic Linear Model

Recall, the univariate DLM is specified by the *observation equation* and *state equation*

$$Y_t = \boldsymbol{X}_t \boldsymbol{\beta}_t + v_t \qquad v_t \sim N(0, \sigma^2) \ ,$$

$$\boldsymbol{\beta}_t = \boldsymbol{G}_t \boldsymbol{\beta}_{t-1} + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim N(\boldsymbol{0}, \boldsymbol{W}) \ ,$$

where Y_t is the observed scalar at time t, $\boldsymbol{\beta}_t$ is the p-dimensional state vector, $\boldsymbol{X}_t$ is a vector of known covariates, and the errors v_t and $\boldsymbol{w}_t$ are two independent sequences of independent Gaussian errors with mean zero and known variance components. We will assume $\boldsymbol{G}_t$ is the identity matrix, which implies each element of the state vector varies according to a random walk.

From this representation, it is easy to see that the DLM satisfies the following assumptions.

- $\boldsymbol{\beta}_t$ is a Markov chain.
- Conditionally on $\boldsymbol{\beta}_t$, the observable Y_t's are independent.

As a result, the DLM is completely specified by the initial distribution $\pi(\boldsymbol{\beta}_0)$ and the conditional densities $\pi(\boldsymbol{\beta}_t|\boldsymbol{\beta}_{t-1})$ and $\pi(y_t|\boldsymbol{\beta}_t)$,

$$\pi(\boldsymbol{\beta}_{0:T}, \boldsymbol{y}_{1:T}) = \pi(\boldsymbol{\beta}_0) \prod_{i=1}^{T} \pi(\boldsymbol{\beta}_i|\boldsymbol{\beta}_{i-1}) \pi(y_i|\boldsymbol{\beta}_i) \ . \tag{3.7}$$

Assuming the initial distribution $\pi(\boldsymbol{\beta}_0)$ and the conditional densities $\pi(\boldsymbol{\beta}_t|\boldsymbol{\beta}_{t-1})$ and $\pi(y_t|\boldsymbol{\beta}_t)$ are all Gaussian, then it can be shown the random vector $(\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_t, Y_1, \ldots, Y_t)$ has a Gaussian distribution for any $t \geqslant 1$. From multivariate normal theory, the marginal and conditional distributions are also Gaussian and therefore, are completely determined by their means and variances. As a result, the filtering distributions can be computed sequentially as new information becomes available using the Kalman filter. The solution of filtering the DLM, or updating the state vector Gaussian distribution, is as follows [Petris et al., 2009]. Begin by letting $\boldsymbol{\beta}_{t-1}|\boldsymbol{y}_{1:t-1} \sim N(\boldsymbol{m}_{t-1}, \boldsymbol{C}_{t-1})$.

- The one step ahead predictive distribution of $\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t-1}$ is Gaussian with parameters

$$\boldsymbol{a}_t = E(\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t-1}) = \boldsymbol{G}_t\boldsymbol{m}_{t-1} \ , \qquad \boldsymbol{R}_t = Var(\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t-1}) = \boldsymbol{G}_t\boldsymbol{C}_{t-1}\boldsymbol{G}'_t + \boldsymbol{W} \ . \tag{3.8}$$

- The one step ahead predictive distribution of $Y_t|\boldsymbol{y}_{1:t-1}$ is Gaussian with parameters

$$f_t = E(Y_t|\boldsymbol{y}_{1:t-1}) = \boldsymbol{X}_t\boldsymbol{a}_t \ , \qquad Q_t = Var(Y_t|\boldsymbol{y}_{1:t-1}) = \boldsymbol{X}_t\boldsymbol{R}_t\boldsymbol{X}'_t + \sigma^2 \ . \tag{3.9}$$

- The filtering distribution of $\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t}$ is Gaussian with parameters

$$\boldsymbol{m}_t = E(\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t}) = \boldsymbol{a}_t + \boldsymbol{R}_t\boldsymbol{X}'_t\boldsymbol{Q}_t^{-1}e_t \ , \ \ \boldsymbol{C}_t = Var(\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t}) = \boldsymbol{R}_t - \boldsymbol{R}_t\boldsymbol{X}'_tQ_t^{-1}\boldsymbol{X}_t\boldsymbol{R}_t \ , \tag{3.10}$$

where $e_t = Y_t - f_t$ is the forecast error.

The Kalman filter detailed above assumes the variance components of the observation equation and state equation are known. Assuming σ^2 and $\boldsymbol{W}$ are unknown, the joint distribution is specified as

$$\pi(\boldsymbol{\beta}_{0:t}, \sigma^2, \boldsymbol{W}, \boldsymbol{y}_{1:t}) = \pi(\boldsymbol{\beta}_0) \times \pi(\boldsymbol{W}) \times \pi(\sigma^2) \times \prod_{i=1}^{t} \pi(\boldsymbol{\beta}_i|\boldsymbol{\beta}_{i-1}, \boldsymbol{W}) \times \pi(y_i|\boldsymbol{\beta}_i, \sigma^2) \ . \tag{3.11}$$

As with most modern Bayesian analyses, it is generally impossible to compute the posterior distribution, $\pi(\boldsymbol{\beta}_{0:T}, \sigma^2, \boldsymbol{W}|\boldsymbol{y}_{1:T})$, in closed form and therefore, MCMC techniques are required.

Applying the Gibbs sampler, one must sample from the full conditional densities of $\pi(\sigma^2|\boldsymbol{\beta}_{0:T}, \boldsymbol{W}, \boldsymbol{y}_{1:T})$, $\pi(\boldsymbol{W}|\boldsymbol{\beta}_{0:T}, \sigma^2, \boldsymbol{y}_{1:T})$, and $\pi(\boldsymbol{\beta}_{0:T}|\sigma^2, \boldsymbol{W}, \boldsymbol{y}_{1:T})$. Sampling from the full conditionals for the variance components is problem specific. To sample from the unobservable states however, Carter and Kohn [1994] developed the Forward Filtering Backward Sampling (FFBS) algorithm to draw from the distribution of all states $\boldsymbol{\beta}_{0:T}$. FFBS is detailed below.

1. Run the Kalman filter and save the Gaussian parameters $\boldsymbol{m}_t$ and $\boldsymbol{C}_t$ $\forall t$.

2. Draw the terminal state $\boldsymbol{\beta}_T \sim N(\boldsymbol{m}_T, \boldsymbol{C}_T)$ first.

3. For $t = T-1, \ldots, 0$ draw $\boldsymbol{\beta}_t \sim N(\boldsymbol{h}_T, \boldsymbol{H}_T)$ where $\boldsymbol{h}_t$ and $\boldsymbol{H}_T$ are computed recursively, similar to the Kalman filter.

    - $\boldsymbol{h}_t = \boldsymbol{m}_t + \boldsymbol{C}_t \boldsymbol{G}'_{t+1} \boldsymbol{R}^{-1}_{t+1}(\boldsymbol{\beta}_{t+1} - \boldsymbol{a}_{t+1}) \qquad \boldsymbol{H}_t = \boldsymbol{C}_t - \boldsymbol{C}_t \boldsymbol{G}'_{t+1} \boldsymbol{R}^{-1}_{t+1} \boldsymbol{G}_{t+1} \boldsymbol{C}_t$

Coupling FFBS and the Gibbs sampler we can draw samples from $\pi(\boldsymbol{\beta}_{0:T}, \sigma^2, \boldsymbol{W}|\boldsymbol{y}_{1:T})$ as follows.

1. Initialize $\sigma^{2(0)}$ and $\boldsymbol{W}^{(0)}$

2. For iterations $i = 1, \ldots, N$:

    (a) Draw $\boldsymbol{\beta}^{(i)}_{0:T}$ from $\pi(\boldsymbol{\beta}_{0:T}|\boldsymbol{y}_{1:T}, \sigma^2 = \sigma^{2(i-1)}, \boldsymbol{W} = \boldsymbol{W}^{(i-1)})$ using FFBS.

    (b) Draw $\pi(\sigma^{2(i)}|\boldsymbol{y}_{1:T}, \boldsymbol{\beta}_{0:T} = \boldsymbol{\beta}^{(i)}_{0:T}, \boldsymbol{W} = \boldsymbol{W}^{(i-1)})$ as necessary.

    (c) Draw $\pi(\boldsymbol{W}^{(i)}|\boldsymbol{y}_{1:T}, \boldsymbol{\beta}_{0:T} = \boldsymbol{\beta}^{(i)}_{0:T}, \sigma^2 = \sigma^{2(i)})$ as necessary.

3.5.2 Fitting the Dynamic Generalized Linear Model

West et al. [1985] coined the term dynamic generalized linear model (DGLM), which is simply an extension of the generalized linear model to time series data. DGLMs maintain the dynamic parameter framework while modeling a sequence of observations from a non-normal sampling distribution. For this chapter we focus solely on the DGLM with a Bernoulli response. Recall, for dynamic logistic regression the *observation equation* is defined as

$$y_t|\boldsymbol{\beta}_t \sim Bernoulli(\pi_t) \ , \quad \pi_t = \frac{e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}}{1+e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}} \quad t = 1, \ldots, T \ , \tag{3.12}$$

and the *state equation* is the same as in the DLM scenario,

$$\boldsymbol{\beta}_t = \boldsymbol{G}_t\boldsymbol{\beta}_{t-1} + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim N(\boldsymbol{0}, \boldsymbol{W}) \ . \tag{3.13}$$

In this case the response $y_t \in \{1, 0\}$ represents a "success" or "failure" respectively at time t and the probability of success at time t is linked to the p-dimensional state vector $\boldsymbol{\beta}_t$ and known vector of covariates $\boldsymbol{X}_t$. The matrices $\boldsymbol{G}_t$ and $\boldsymbol{W}$ maintain the same purpose and interpretation as before.

In the case of DLM's, the one step ahead predictive distribution of $\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t-1}$, the one step ahead predictive distribution of $Y_t|\boldsymbol{y}_{1:t-1}$, and the filtering distribution of $\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t}$ are all calculated online because their Gaussian parameters are completely and sequentially determined by the Kalman filter. The same technique is, of course, not possible in the DGLM framework.

However, a variety of techniques have been proposed to sequentially update the state vector $\boldsymbol{\beta}_t$. For example, West et al. [1985] describe an approach using Linear Bayes estimation. The authors make no assumption regarding the distribution of the state equation (Equation 3.13), only that the first and second order moments are defined, $\boldsymbol{w}_t \sim (\boldsymbol{0}, \boldsymbol{W})$. The goal is to filter the distribution of the state vector $\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t-1} \sim (\boldsymbol{m}_{t-1}, \boldsymbol{C}_{t-1})$ to $\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t} \sim (\boldsymbol{m}_t, \boldsymbol{C}_t)$ once a new observation becomes available. The authors proceed to update the state vector similarly to the DLM of Section 3.5.1, avoiding distributional assumptions. As a result, the technique requires various iterative approximations. For a complete description of the method, see West et al. [1985].

Other methods to sequentially update the state vector proposed since West et al. [1985] generally rely on linearly approximating the observation equation to allow for the assumption of normality. The state vector can than be sequentially updated using the Kalman filter. For additional details and information on this technique and others, see Ferreira and Gamerman [2000].

Moving away from sequential approximation, more modern DGLM fitting techniques rely on MCMC and can be applied in a similar fashion as the DLM. The joint posterior density for the general exponential family dynamic model is

$$\pi(\boldsymbol{y}_{1:T}, \boldsymbol{\beta}_{0:T}, \boldsymbol{W}) = \pi(\boldsymbol{\beta}_0) \times \pi(\boldsymbol{W}) \times \prod_{t=1}^{T} \pi(y_t|\boldsymbol{\beta}_t) \times \prod_{t=1}^{T} \pi(\boldsymbol{\beta}_t|\boldsymbol{\beta}_{t-1}, \boldsymbol{W}) \ . \qquad (3.14)$$

Applying the Gibbs sampler, one must sample from the full conditional densities $\pi(\boldsymbol{W}|\boldsymbol{\beta}_{0:T}, \boldsymbol{y}_{1:T})$ and $\pi(\boldsymbol{\beta}_{0:T}|\boldsymbol{W}, \boldsymbol{y}_{1:T})$. Sampling from the full conditional for the evolution matrix is problem specific. Sampling from the states is accomplished via the Metropolis-Hastings algorithm within the Gibbs sampler using a pseudo FFBS algorithm. The MCMC approach is detailed below [Gamerman, 1998].

1. Initialize the matrices $\boldsymbol{W}^{(0)}$ and $\boldsymbol{\beta}_{0:T}^{(0)}$.

2. For iterations $i = 1, \ldots, N$:

   (a) Draw elements of $\boldsymbol{\beta}_{0:T}^{(i)}$ component by component.

   That is, draw $\boldsymbol{\beta}_t^{(i)}$ from $\pi(\boldsymbol{\beta}_t | \boldsymbol{\beta}_{0:(t-1)}^{(i)}, \boldsymbol{\beta}_{(t+1):T}^{(i-1)}, \boldsymbol{y}_{1:T}, \boldsymbol{W} = \boldsymbol{W}^{(i-1)})$.

   (b) Draw $\pi(\boldsymbol{W}^{(i)} | \boldsymbol{y}_{1:T}, \boldsymbol{\beta}_{0:T} = \boldsymbol{\beta}_{0:T}^{(i)})$ as necessary.

Gamerman [1998] suggests using the conjugate inverted Wishart prior distribution for $\boldsymbol{W}$. Thus, the full conditional distribution for $\boldsymbol{W}$ is also an inverted Wishart distribution making it easy to sample the $p \times p$ matrix at each iteration.

One drawback of the MCMC approach is that convergence can be slow [Carter and Kohn, 1994]. Sampling each element of the state vector for each time period is burdensome as the number of states grows linearly with time, $T \times p$. A few approaches have been proposed to speed up the simulation. Most approaches tweak the proposal values within the Metropolis-Hastings algorithm and simulate states in blocks, such as $(\boldsymbol{\beta}_r, \ldots, \boldsymbol{\beta}_s)$ where $1 \leqslant r < s \leqslant T$ [Ferreira and Gamerman, 2000].

3.6 Model

Here we describe the dynamic logistic regression model fitting approach and the variable selection technique proposed for a non-stationary binary time series. We first describe the latent variable augmentation approach used to more efficiently fit the dynamic logistic regression model and then the penalized credible region variable selection technique that can be extended to dynamic models.

3.6.1 Model Fit

In the static logistic regression scenario, Polson et al. [2013] proposed a data augmentation technique for fully Bayesian inference when considering binomial likelihoods. The authors introduce the Pólya-Gamma random variable to be used as a latent variable in Bayesian analyses. The random variable X has a Pólya-Gamma distribution, denoted $X \sim PG(b, c)$, if

$$X = \frac{1}{2\pi^2} \sum_{k=1}^{inf} \frac{g_k}{(k - 1/2)^2 + c^2/4\pi^2} \ ,$$

where $b > 0$, $c \in \mathbb{R}$, $g_k \sim Gamma(b, 1)$ are independent Gamma random variables. Simulated values from the Pólya-Gamma distribution can be generated using the R package `BayesLogit`.

The main result from Polson et al. [2013] is that likelihoods from binomial response data can be represented as mixtures of Gaussians according to the Pólya-Gamma distribution,

$$\frac{exp(\psi)^a}{(1 + exp(\psi))^b} = \frac{exp(k\psi)}{2^b} \int_0^{inf} exp(-\omega\psi^2/2)p(\omega)d\omega \ ,$$

where $k = a - b/2$ and $\omega \sim PG(b, 0)$. Replacing ψ with $\boldsymbol{X\beta}$ presents a Gaussian kernel in $\boldsymbol{\beta}$ above. Also, the conditional distribution for $\omega|\psi$ is a Pólya-Gamma distribution, ideal for obtaining posterior draws.

The addition of the latent variable allows for an efficient Gibbs sampler. The authors show that to sample from the posterior distribution in the static logistic regression model with a Gaussian prior for $\boldsymbol{\beta}$, simply sample from the following full conditional distributions,

$$\pi(\boldsymbol{\omega}|\boldsymbol{\beta}) \sim PG(1, \boldsymbol{X}\boldsymbol{\beta}) \qquad \text{and} \qquad \pi(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{\omega}) \sim N(\boldsymbol{m}_\omega, \boldsymbol{V}_\omega) \ ,$$

where the full conditional for $\boldsymbol{\beta}$ is multivariate normal with parameters which depend on ω.

Windle et al. [2013] extend the Pólya-Gamma data augmentation approach to dynamic logistic regression. Starting with the likelihood of observed data $\boldsymbol{y}_{1:T}$ and distribution of hidden states $\boldsymbol{\beta}_{1:T}$,

$$\pi(\boldsymbol{\beta}_{1:T}|\boldsymbol{y}_{1:T}) = \left[\prod_{t=1}^{T} \frac{exp(\boldsymbol{X}_t\boldsymbol{\beta}_t)^{y_t}}{1 + exp(\boldsymbol{X}_t\boldsymbol{\beta}_t)}\right]\pi(\boldsymbol{\beta}_{1:T}) \ , \tag{3.15}$$

the authors introduce the Pólya-Gamma latent variable, $\omega_t \sim PG(1, \psi_t)$, for $t = 1, \ldots, T$ to construct the joint distribution,

$$\pi(\boldsymbol{\beta}_{1:T}, \boldsymbol{\omega}_{1:T}|\boldsymbol{y}_{1:T}) = \left[\prod_{t=1}^{T} \frac{exp(\boldsymbol{X}_t\boldsymbol{\beta}_t)^{y_t}}{1 + exp(\boldsymbol{X}_t\boldsymbol{\beta}_t)}\pi(\omega_t|\boldsymbol{X}_t\boldsymbol{\beta}_t)\right]\pi(\boldsymbol{\beta}_{1:T}) \ . \tag{3.16}$$

Using properties of the Pólya-Gamma distribution, the joint posterior can be rewritten as

$$\pi(\boldsymbol{\beta}_{1:T}, \boldsymbol{\omega}_{1:T}|\boldsymbol{y}_{1:T}) \ \propto \ \left[\prod_{t=1}^{T} exp\bigg(- \frac{\omega_t}{2}\Big(\frac{k_t}{\omega_t} - \boldsymbol{X}_t\boldsymbol{\beta}_t\Big)^2\bigg)\right]\pi(\boldsymbol{\beta}_{1:T}) \ , \tag{3.17}$$

where $k_t = y_t - 1/2$. The addition of latent variable provides pseudo data $z_t = k_t/\omega_t$ where $z_t \sim N(\boldsymbol{X}_t\boldsymbol{\beta}_t, 1/\omega_t)$. If we specify $\pi(\boldsymbol{\beta}_{1:T})$ such that the states vary according to a random walk (Equation 3.13), then sampling from the conditional distribution for $\boldsymbol{\beta}_{1:T}$ is equivalent to sampling from the DLM *observation equation* and *state equation*,

$$z_t = \boldsymbol{X}_t\boldsymbol{\beta}_t + v_t \qquad v_t \sim N(0, 1/\omega_t) \ , \tag{3.18a}$$

$$\boldsymbol{\beta}_t = \boldsymbol{G}_t \boldsymbol{\beta}_{t-1} + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim N(\boldsymbol{0}, \boldsymbol{W}) \; . \tag{3.18b}$$

As a result, the posterior simulation can now be implemented using the Forward Filtering Backward Sampling (FFBS) algorithm to sample state vectors as described in Section 3.5.1. The combination of the Pólya-Gamma latent variable augmentation approach and FFBS is highly advantageous and more computationally efficient than other DGLM fitting methods because it avoids analytic approximations, numerical integration, and use of the Metropolis-Hastings algorithm. The full posterior simulation for dynamic logistic regression is detailed below.

The joint posterior density for the general exponential family dynamic model from Equation (3.14), is now written to include the latent variable,

$$\pi(\boldsymbol{y}_{1:T}, \boldsymbol{\beta}_{0:T}, \boldsymbol{W}, \boldsymbol{\omega}_{1:T}) = \pi(\boldsymbol{\beta}_0) \times \pi(\boldsymbol{W}) \times \prod_{t=1}^{T} \pi(y_t|\boldsymbol{\beta}_t) \times \prod_{t=1}^{T} \pi(\boldsymbol{\beta}_t|\boldsymbol{\beta}_{t-1}, \boldsymbol{W}) \prod_{t=1}^{T} \pi(\omega_t|\boldsymbol{\beta}_{t-1}). \tag{3.19}$$

For dynamic logistic regression, specify $\pi(y_t|\boldsymbol{\beta}_t)$ as the Bernoulli observation equation, $\pi(\boldsymbol{\beta}_t|\boldsymbol{\beta}_{t-1}, \boldsymbol{W})$ as the normally distributed state equation, and $\pi(\omega_t|\boldsymbol{\beta}_{t-1})$ as the Pólya-Gamma distributed latent variable. Let $\pi(\boldsymbol{\beta}_0)$ be the normally distributed prior for the initial state vector, which is now a conjugate prior due to the latent variable, and let $\pi(\boldsymbol{W})$ be the prior for the state equation covariance. For computational purposes, assume the covariance structure is $\boldsymbol{W} = diag\left(\frac{1}{\tau_1}, \ldots, \frac{1}{\tau_p}\right)$, where τ_i represents the i^{th} inverse variance component. Therefore, the prior can be written as a product due to its diagonal form, $\prod_{i=1}^{p} \pi(\tau_i)$. Let the inverse variance components be gamma distributed, $\pi(\tau_i) \sim gamma(\alpha, \gamma)$, to provide a conjugate prior. Given that the posterior density is completely specified, now draw posterior samples of $\{\boldsymbol{\beta}_{1:T}, \boldsymbol{W}, \boldsymbol{\omega}_{1:T}\}$ from the joint posterior distribution in Equation (3.19).

1. Initialize the latent variable vector $\boldsymbol{\omega}_{1:T}^{(0)}$, the states $\boldsymbol{\beta}_{1:T}^{(0)}$, and the state covariance $\boldsymbol{W}^{(0)}$.

2. For iterations $k = 1, \ldots, N$:

   (a) Sample $\boldsymbol{\beta}_{1:T}^{(k)}$ using the FFBS algorithm using pseudo Gaussian data from Equation 3.18a.

   (b) Sample the components of $\boldsymbol{W}^{(k)}$ individually from the updated Gamma distribution,

   $$\pi\big(\tau_i^{(k)} | \cdot\big) \sim Gamma\Bigg(\alpha + \frac{T}{2}, \gamma + \sum_{t=1}^{T}(\beta_{ti}^{(k)} - \beta_{(t-1)i}^{(k)})^2\Bigg) ,$$

   for $i = 1, \ldots, p$.

   (c) Sample each element of the latent variable vector $\omega_t^{(k)}$, from the Pólya-Gamma distribution conditioned on the states, $PG(1, \boldsymbol{X}_t \boldsymbol{\beta}_t^{(k)})$, for $t = 1, \ldots, T$.

3.6.2 Variable Selection

An ideal variable selection method for dynamic models is one that can scale to models with many predictors and can efficiently conduct variable selection each time the model is updated. Here we describe a method originally developed for Bayesian linear models and show that it can be extended to dynamic models.

Bondell and Reich [2012] proposed Bayesian variable selection via penalized credible regions for the traditional linear model (Equation (3.3)). This approach separates the model fitting and variable selection process by constructing credible regions from the posterior distribution of $\boldsymbol{\beta}$ and Σ. Given a $(1-\alpha)\times 100\%$ credible region, any point within the region is a feasible estimate of $\boldsymbol{\beta}$. The authors suggest selecting the sparsest solution to accomplish variable selection. Thus, the proposed estimate is

$$\tilde{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\boldsymbol{\beta}\|_0 \qquad \text{subject to } \boldsymbol{\beta} \in C_\alpha \ , \tag{3.20}$$

where $\|\boldsymbol{\beta}\|_0$ is the L_0 norm of the vector $\boldsymbol{\beta}$ and C_α is the $(1-\alpha)\times 100\%$ credible region. The selected model excludes predictors where $\beta_j = 0$ is included in the credible region.

As the coverage increases, the credible region expands leading to a more sparse model. The authors suggest creating a sequence of credible sets that correspond to creating a sequence of selected models. The sequence of p models are the only p models to be considered. The best model, given the solution path, can then be chosen by a goodness-of-fit metric such as AIC or BIC. This is a drastic reduction in the comparisons required for all possible 2^p models.

Figure 3.1 demonstrates the joint credible region approach to variable selection for a linear model with only two parameters, β_1 and β_2. Starting from the largest ellipse, referring to the 95% credible region, the sparsest solution would be the null model as $\beta_1 = \beta_2 = 0$ is a feasible solution. Next, the 90% credible region indicates only $\beta_2 = 0$. Finally, the sparsest solution within the smallest credible region includes both parameters in the model as nonzero coefficients. Therefore, the solution path in this example is $\{\beta_1, \beta_2\}$.

In some cases, the posterior of $\boldsymbol{\beta}$ is elliptical with density of the form $H[(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \hat{\Sigma}^{-1} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})]$ where H is a monotone decreasing function. Therefore, the highest density region is of the form $\{(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \hat{\Sigma}^{-1} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) = K_\alpha\}$ for some K_α. In general, the posterior distribution does not maintain elliptical contours but credible sets can still be

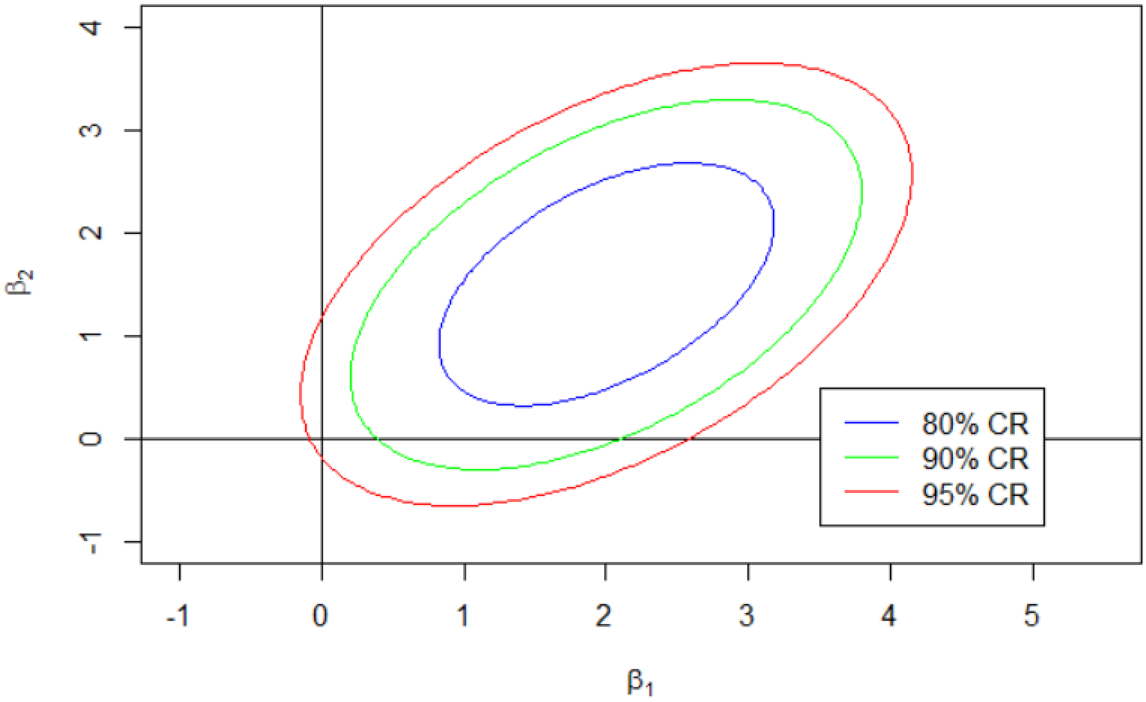


Figure 3.1: Example of the joint credible region variable selection approach for a linear model with two parameters. Here the solution path is $\{\beta_1 . \beta_2\}$.

created. In addition, the solution to Equation (3.20) is not unique and requires searching over a possibly high dimensional region. To overcome these challenges, Bondell and Reich [2012] apply several alterations to Equation (3.20). First, the authors replace the L_0 norm by a smooth combination of L_0 and L_1 [Lv and Fan, 2009] which leads to a non-convex optimization problem. Next, it is converted to a convex optimization problem by applying a local linear approximation which is equivalent to the Lagrangian optimization problem

$$\tilde{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_j|^2} \,, \tag{3.21}$$

where λ is the tuning parameter and has one-to-one correspondence to α. Therefore, the proposed sequence of selected models is given by the solution to Equation 3.21 as a function of λ. For a given λ, regressing $\boldsymbol{Y}^* = \hat{\boldsymbol{\Sigma}}^{-1/2}\hat{\boldsymbol{\beta}}$ on $\boldsymbol{X}^* = \hat{\boldsymbol{\Sigma}}^{-1/2}\boldsymbol{D}$ with an L_1 penalization, where $\boldsymbol{D} = diag(\hat{\beta}_1^{\;2}, \ldots, \hat{\beta}_p^{\;2})$, returns the solution $\boldsymbol{\beta}^*$. The solution to Equation (3.21) is then calculated as $\tilde{\boldsymbol{\beta}} = \boldsymbol{D}\boldsymbol{\beta}^*$. One can readily find the entire solution path using the LARS algorithm [Efron et al., 2004].

The joint credible region variable selection approach can be extended to the dynamic logistic regression model by altering the inputs to the optimization problem in Equation (3.21). In order to create the solution path for a linear model detailed above, the technique only requires posterior distribution estimates of the parameter vector $\hat{\boldsymbol{\beta}}$ and model covariance $\hat{\Sigma}$. To extend this approach to dynamic logistic regression we can replace these inputs with posterior estimates of the state vector $\hat{\boldsymbol{\beta}}_t$ and its covariance $\hat{\boldsymbol{W}}$. Thus, the optimization problem becomes

$$\tilde{\boldsymbol{\beta}}_t = \underset{\boldsymbol{\beta}_t}{\operatorname{argmin}}(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_t)^T \hat{\boldsymbol{W}}^{-1}(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_t) + \lambda \sum_{j=1}^{p} \frac{|\beta_{jt}|}{|\hat{\beta_{jt}}|^2} \; . \tag{3.22}$$

There are two advantages of the joint credible region variable selection approach. First, this method is scalable because we are able to include all p predictors into the model fitting process, unlike the dynamic model averaging approach in McCormick et al. [2012]. Second, this method allows for dynamic variable selection, meaning, we are able to construct joint credible regions and conduct variable selection at every time point for every state vector $\hat{\boldsymbol{\beta}}_t$. As new information becomes available and the underlying process generating the model changes, the dynamic joint credible region approach can determine nonzero elements of the state vector sequentially in time.

3.7 Simulation

In this section, we evaluate the performance of the dynamic logistic regression and variable selection approach of Section 3.6 against the static logistic regression with LASSO regularization model. The latter approach is used in the motivating example to predict civil unrest in South America [Korkmaz et al., 2015]. We compare both prediction and variable selection results for each model.

3.7.1 Setup

In each case of the simulation, data sets are generated from the dynamic logistic regression *observation equation*

$$y_t|\boldsymbol{\beta}_t \sim Bernoulli(\pi_t)\ , \quad \pi_t = \frac{e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}}{1+e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}} \quad t = 1,\ldots,T\ ,$$

and *state equation*

$$\boldsymbol{\beta}_t = \boldsymbol{G}_t\boldsymbol{\beta}_{t-1} + \boldsymbol{w}_t\ , \qquad \boldsymbol{w}_t \sim N(\boldsymbol{0}, \boldsymbol{W})\ .$$

The number of observations for each time series is varied in $T \in \{50, 100\}$ and the covariates are standard normal. We let the parameters freely vary according to a random walk by setting both the transition matrix and the state equation covariance matrix to the identity, $\boldsymbol{G}_t = \boldsymbol{W} = \boldsymbol{I}_p$. The number of candidate predictors is varied in $p \in \{25, 50, 100\}$ and the true number of predictors used to generate $\boldsymbol{y}_{1:T}$ is approximately ten percent of of p, $p^* \in \{3, 5, 10\}$. For each data set, p^* indices are randomly chosen as the nonzero locations of the state vectors $\boldsymbol{\beta}_t$ to avoid a potential ordering bias.

To simulate the nonzero elements of the state vectors we consider two scenarios for generating dynamic coefficients: structural break parameters and completely dynamic parameters. First, completely dynamic parameters are simulated as seen in the state equation above. That is, the nonzero parameters vary according to a random walk by adding white noise to those parameters at every single time point. Structural break parameters are a relaxed version of the completely dynamic scenario. In this case, parameters are assumed static for $t > 1$ until the model experiences a shift. For this simulation, we shock the true p^* parameters every 10 time points by simply adding white noise to the parameters. For structural break parameters the state equation is written

$$\boldsymbol{\beta}_t = \boldsymbol{G}_t \boldsymbol{\beta}_{t-1} + \boldsymbol{w}_t \mathbb{1}\big(0 \equiv t \mod 10\big) \qquad \boldsymbol{w}_t \sim N(\mathbf{0}, \boldsymbol{W}) \ .$$

Figure 3.2 shows how a single parameter may evolve in time under the two scenarios. Clearly, it should be easier for the model to track strucutral break parameters as there are periods of stationarity and the parameters lack extreme movement during the shock. The completely dynamic parameter scenario adds an increased layer of complexity. For example, Figure 3.2 shows that the completely dynamic parameter is negative at first and then increases to a relatively large positive coefficient until it moves back to zero at the end of the graph. The goal is to see if adding complexity to the model reduces performance given that it is more difficult to track completely dynamic parameters.

3.7.2 Metrics

For each time series length $T \in \{50, 100\}$, we fit the model on the first 50% of the observations, $\{25, 50\}$, and then forecast one time period ahead and five time periods ahead. For the LASSO regularization of the static model, the regularization parameter is chosen via five-fold cross-validation. For the dynamic logistic regression variable selection,

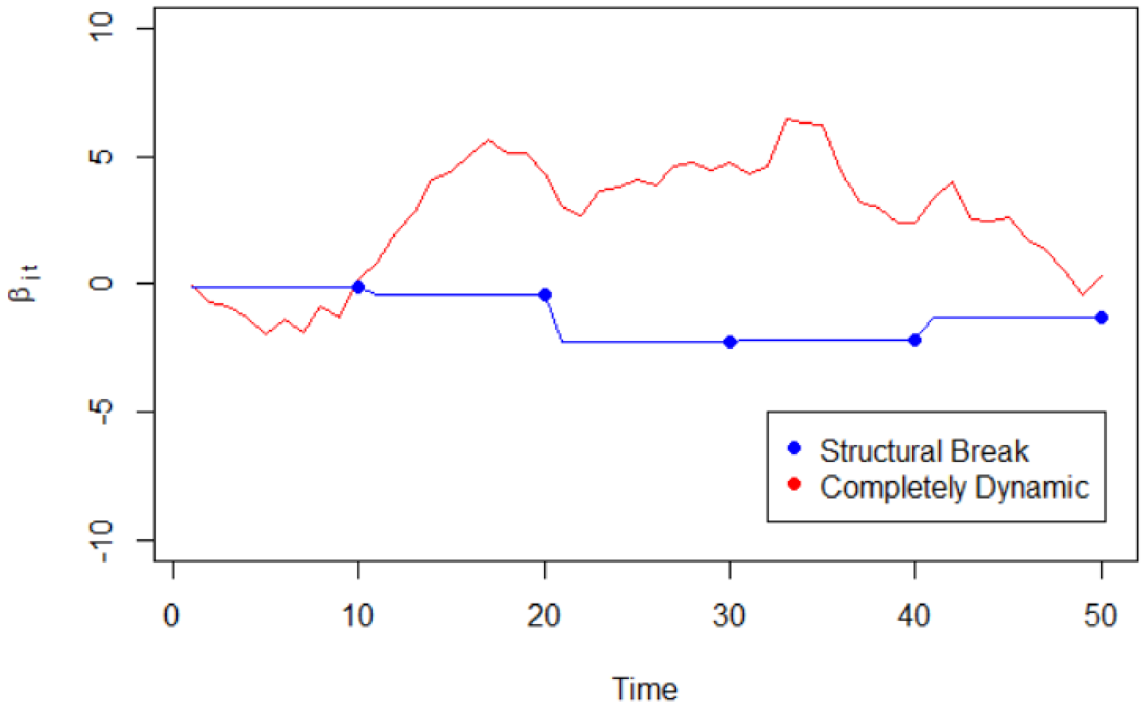


Figure 3.2: Example of a single structural break and completely dynamic parameter.

we apply the penalized credible region approach of Section 3.6.2 on the terminal state vector. We then move one unit of time ahead, fit the model to the past data, forecast, and conduct variable selection again. The process is repeated until the end of the times series. At each forecast and variable selection period, we measure the accuracy of predictions with the forthcoming metrics.

Prediction performance is measured by the standard confusion matrix for a binary response. Let true positives (TP) be a correctly predicted event, true negatives (TN) a correctly predicted nonevent, false positives (FP) an incorrectly predicted event, and false negatives (FN) an incorrectly predicted nonevent. Overall model accuracy can be measured as the proportion of correct predictions, $\frac{TP+TN}{TP+TN+FP+FN}$. For each of the four prediction performance possibilities we use a cutoff rate of 0.50.

We also examine the trade-off between power and the false discovery rate (FDR) by measuring both recall and precision for each model. Recall, also known as the true positive rate (TPR), is the ratio of true positives to the total number of events, $\frac{TP}{TP+FN}$. Hence, it measures the ability for the model to correctly predict an event when there is in fact an event. Precision on the other hand, is the ratio of true positives to the total number of predicted events, $\frac{TP}{TP+FP}$. These two metrics must be measured in combination because, for example, recall could be inflated by allowing the model to simply predict events in all cases. To aggregate precision and recall, we report the F1 scores, or the harmonic average of precision and recall, for each case in the simulation.

Variable selection performance is measured using the same metrics as prediction. For each prediction, we count the number of predictors correctly included in the model (TP), correctly excluded from the model (TN), incorrectly included (FP), and incorrectly excluded (FN). In addition, the precision, recall, and F1 scores for variable selection are also measured for the same reasons above.

3.7.3 Results

For each combination of time series length $T \in \{50, 100\}$, candidate predictors $p \in \{25, 50, 100\}$, and dynamic parameter scenario, we average the metrics across 100 data sets. For each data set we use 5000 MCMC iterations to approximate the posterior distribution after a 500 iteration burn-in period. Results for structural break parameters are displayed in Tables 3.1-3.3 and completely dynamic parameter results are in Tables 3.4-3.6.

For the prediction results from structural break parameters, perhaps the first thing to notice is the accuracy of the dynamic logistic regression model is much higher than the LASSO model. That is, the true positive and true negative rates for the DGLM are close to 50% each, and therefore, false predictions are extremely low. For the LASSO model, it appears that a majority of the predictions are events as the true positive and false positive rates are near to 50% each. As a result, the LASSO model rarely makes the correct prediction of a nonevent.

Due to the high rate of false positives for the LASSO model, the precision is held low ranging between $50\% - 53\%$. On the other hand, recall is high in the upper 90% range for the LASSO model because the false negative rate is low. The low precision and high recall again indicate that the LASSO model over predicts the rate of events in each data set. Averaging out the low precision and high recall for the LASSO model gives F1 scores in the $67\% - 69\%$ range. Due to the high accuracy of the DGLM model the precision, recall, and F1 scores are all in the range of $95\% - 100\%$.

Moving from a times series length of $T = 50$ in Table 3.1 to $T = 100$ in Table 3.2 appears to make no impact on prediction results for the DGLM model, either forecasting one time period ahead or five periods ahead. F1 scores remain high in either case. The same is true when T is held constant and the number of candidate predictors varies in $p = \{25, 50, 100\}$. F1 scores remain high even moving into the $p > T$ scenario.

The variable selection for all cases of structural break parameters (Table 3.3) is more competitive than prediction performance between the two models. However, in all cases, the true positive rate for the DGLM is higher than that of the LASSO and the false negative rate is also lower. The DGLM more accurately includes the correct variables into the model. In terms of the true negative rate and false positive rate, the LASSO model slightly outperforms the DGLM in all cases except $T = p = 50$. For structural break parameters, the LASSO model more accurately excludes noise predictors. Furthermore, both models share that the true positive rate and the F1 scores increase as the number of observations increase from $T = 50$ to $T = 100$, holding p constant. Intuitively this makes sense, the more data each model has, the better they are equipped to pick the correct variables. Finally, in all cases of Table 3.3 the precision, recall, and F1 scores are all higher for the DGLM. On average the DGLM is better able to pick out the true predictors and reject the noise variables.

Note, the variable selection performance values in Table 3.3 give the impression of weak performance compared to the prediction results in Tables 3.1 and 3.2. The seemingly low values for the true positive rate, precision, recall, and F1 scores in Table 3.3 are due to the simulation setup. For the candidate predictors $p = \{25, 50, 100\}$, the true number of predictors is $p^* = \{3, 5, 10\}$ and therefore, the maximum true positive rates are only $\{12\%, 10\%, 10\%\}$ for each case. Consequently, the true positive rate will always remain low. The false predictions are near equivalent values of the true positive rate giving low precision, recall, and F1 scores. Regardless of these values, the simulation still shows that the DGLM outperforms the LASSO model in all cases.

The results for the completely dynamic parameter scenario (Tables 3.4-3.6) are similar to the structural break parameter scenario (Tables 3.1-3.3). For prediction, the same pattern emerges, the accuracy of the DGLM far superior than that of the LASSO model. The precision for the LASSO hovers around the 50% mark, the recall is in the high 90% range, and the F1 scores are between $66\% - 69\%$. Again, the precision, recall and F1 scores for the DGLM outperforms the LASSO model in every instance with values between $97\% - 100\%$.

In terms of variable selection for the completely dynamic parameter scenario in Table 3.6, the DGLM outperforms the LASSO model in all four categories of the confusion matrix (TP,TN,FP,FN). Thus, moving to the completely dynamic parameter scenario, the true negative rate decreases and the false positive rate increases for the LASSO model.

Furthermore, the precision, recall and F1 scores for variable selection are all higher for the DGLM. The F1 scores for the LASSO and DGLM are in ranges of $21\% - 44\%$ and $31\% - 55\%$ respectively.

In all simulation cases the precision, recall, and F1 scores for the dynamic logistic regression model outperform the LASSO model in both prediction and variable selection. For prediction, the LASSO model consistently over classifies predictions as events and the DGLM was consistently accurate with F1 scores in the high 90% range. For the parameter generating scenarios, prediction results for the LASSO model were similar, but the variable selection results declined. That is, the LASSO model did slightly better for structural break parameters because this scenario is closer to a static model. Regardless of the parameter generating scenario, the DGLM model was better able to track the underlying distribution allowing for better rates of inclusion of true predictors and exclusion of noise variables.

Table 3.1: Forecast results for structural break parameters and $T = 50$.

Model	Forecast	Predictors	TP	TN	FP	FN	Precision	Recall	F1
LASSO	1	25	49.73	1.60	47.97	0.70	50.90	98.61	67.14
	5	25	49.45	1.85	47.98	0.73	50.75	98.54	67.00
DGLM	1	25	50.13	49.10	0.47	0.30	99.07	99.41	99.24
	5	25	48.38	47.60	2.22	1.79	95.61	96.43	96.02
LASSO	1	50	50.43	1.93	46.53	1.10	52.01	97.87	67.92
	5	50	50.60	1.90	46.51	0.98	52.11	98.10	68.06
DGLM	1	50	51.40	48.40	0.06	0.13	99.88	99.74	99.82
	5	50	50.70	47.51	0.90	0.88	98.26	98.29	98.27
LASSO	1	100	50.73	1.33	47.33	0.60	51.73	98.83	67.92
	5	100	50.71	1.04	47.72	0.53	51.51	98.96	67.76
DGLM	1	100	51.23	48.43	0.23	0.10	99.55	99.81	99.68
	5	100	50.61	48.04	0.72	0.63	98.59	98.77	98.68

Table 3.2: Forecast results for structural break parameters and $T = 100$.

Model	Forecast	Predictors	TP	TN	FP	FN	Precision	Recall	F1
LASSO	1	25	51.98	0.62	47.16	0.24	52.43	99.54	68.68
	5	25	51.91	0.63	47.21	0.24	52.37	99.54	68.63
DGLM	1	25	51.60	47.28	0.50	0.62	99.04	98.81	98.92
	5	25	49.60	45.25	2.59	2.56	95.04	95.09	95.06
LASSO	1	50	50.40	0.96	48.24	0.40	51.09	99.21	67.45
	5	50	50.47	1.03	48.16	0.34	51.17	99.33	67.54
DGLM	1	50	50.74	49.16	0.04	0.06	99.92	99.88	99.90
	5	50	49.95	48.50	0.69	0.85	98.64	98.33	98.48
LASSO	1	100	51.60	0.68	47.32	0.40	52.16	99.23	68.38
	5	100	51.67	0.70	47.26	0.37	52.22	99.29	68.45
DGLM	1	100	51.96	48.00	0.02	0.02	99.96	99.96	99.96
	5	100	51.86	47.61	0.35	0.18	99.32	99.65	99.49

Table 3.3: Variable selection results for structural break parameters.

Model	Predictors	T	TP	TN	FP	FN	Precision	Recall	F1
LASSO	25	50	6.02	68.07	19.93	5.98	23.19	50.17	31.73
DGLM	25	50	7.81	67.36	20.64	4.19	27.45	65.08	38.62
LASSO	25	100	8.70	69.22	18.78	3.29	31.66	72.56	44.08
DGLM	25	100	10.62	68.42	19.58	1.37	35.17	88.57	50.34
LASSO	50	50	3.65	77.69	12.31	6.35	22.87	36.50	28.12
DGLM	50	50	4.79	78.68	11.32	5.21	29.73	47.90	36.69
LASSO	50	100	5.95	73.99	16.00	4.05	27.12	59.50	37.24
DGLM	50	100	7.92	72.21	17.79	2.07	30.81	79.28	44.37
LASSO	100	50	1.87	82.91	7.09	8.13	20.87	18.70	19.73
DGLM	100	50	2.48	82.22	7.78	7.52	24.17	24.80	24.48
LASSO	100	100	3.61	80.28	9.71	6.39	27.10	36.10	30.96
DGLM	100	100	5.39	80.05	9.95	4.61	35.14	53.90	42.54

Table 3.4: Forecast results for completely dynamic parameters and $T = 50$.

Model	Forecast	Predictors	TP	TN	FP	FN	Precision	Recall	F1
LASSO	1	25	50.13	2.83	46.60	0.43	51.82	99.14	68.07
	5	25	50.31	2.98	46.15	0.56	52.16	98.89	68.29
DGLM	1	25	50.46	49.33	0.10	0.10	99.80	99.80	99.80
	5	25	49.65	48.16	0.90	1.21	98.22	97.62	97.92
LASSO	1	50	49.23	1.93	48.23	0.60	50.51	98.79	66.84
	5	50	49.29	2.26	47.85	0.59	50.74	98.82	67.05
DGLM	1	50	49.73	50.07	0.10	0.10	99.79	99.79	99.79
	5	50	49.38	49.64	0.48	0.50	99.03	98.99	99.02
LASSO	1	100	50.20	1.30	47.83	0.67	51.21	98.68	67.43
	5	100	50.23	1.57	47.46	0.74	51.41	98.54	67.58
DGLM	1	100	50.87	49.13	0.00	0.00	100.00	100.00	100.00
	5	100	50.83	48.91	0.12	0.14	99.76	99.73	99.74

Table 3.5: Forecast results for completely dynamic parameters and $T = 100$.

Model	Forecast	Predictors	TP	TN	FP	FN	Precision	Recall	F1
LASSO	1	25	50.54	2.34	47.02	0.10	51.80	99.80	68.21
	5	25	50.48	2.29	47.10	0.13	51.73	99.74	68.13
DGLM	1	25	50.62	49.28	0.08	0.02	99.84	99.96	99.90
	5	25	50.01	48.68	0.71	0.51	98.60	98.99	98.79
LASSO	1	50	48.64	2.90	48.36	0.10	50.14	99.79	66.75
	5	50	48.65	2.80	48.37	0.17	50.14	99.65	66.72
DGLM	1	50	48.72	51.26	0.00	0.02	1.00	99.96	99.97
	5	50	48.64	50.99	0.19	0.18	99.61	99.63	99.62
LASSO	1	100	49.40	2.02	48.36	0.22	50.53	99.56	67.04
	5	100	49.53	2.01	48.27	0.19	50.64	99.62	67.15
DGLM	1	100	49.62	50.38	0.00	0.00	100.00	100.00	100.00
	5	100	49.72	50.22	0.06	0.04	99.88	99.92	99.89

Table 3.6: Variable selection results for completely dynamic parameters.

Model	Predictors	T	TP	TN	FP	FN	Precision	Recall	F1
LASSO	25	50	7.16	67.97	20.03	4.84	26.33	59.67	36.54
DGLM	25	50	9.12	75.77	12.23	2.88	42.72	76.00	54.69
LASSO	25	100	8.65	68.91	19.09	3.35	31.18	72.08	43.53
DGLM	25	100	11.29	69.64	18.36	0.70	38.08	94.16	54.23
LASSO	50	50	4.06	76.83	13.17	5.94	23.56	40.60	29.82
DGLM	50	50	5.77	79.95	10.05	4.22	36.47	57.76	44.71
LASSO	50	100	6.66	72.85	17.14	3.33	27.98	66.67	39.42
DGLM	50	100	8.26	76.30	13.69	1.73	37.63	82.68	51.72
LASSO	100	50	2.18	82.08	7.92	7.82	21.58	21.80	21.69
DGLM	100	50	3.07	83.71	6.29	6.92	32.79	30.73	31.73
LASSO	100	100	4.63	77.99	12.00	5.37	27.84	46.30	34.77
DGLM	100	100	5.86	79.04	10.96	4.14	34.84	58.60	43.69

3.8 Application

In this section we first introduce the application of civil unrest and the utility of using daily word counts from Twitter as model features. We then forecast civil unrest and compare results with the proposed dynamic model and static baseline approach using the precision, recall, and F1 metrics of Section 3.7. Variable selection is used to contextualize the reasons for protest and which predictive terms must be actively monitored from Twitter as leading indicators of civil unrest.

3.8.1 Civil Unrest

In recent years open source data has become a powerful tool for predicting real world affairs. The widespread adoption of the social networking site Twitter has resulted in massive repositories of free information. Researchers mine the user generated content and use daily word counts to forecast diverse outcomes. For example, terms like “flu” and “fever” are leading indicators of influenza like illness rates reported by the Centers for Disease Control and Prevention in the United States [Achrekar et al., 2012; Li and Cardie, 2013; Culotta, 2010]. Similarly, daily word counts have been used to predict stock price movement [Bollen et al., 2011], box office revenue for the film industry [Asur and Huberman, 2010], and inner city crime rates [Gerber, 2014]. A summary of current prediction methods and applications using Twitter is described by Arias et al. [2013].

One emerging application for prediction via open source data is civil unrest. A nation’s citizens may protest for a myriad of reasons, from local economic conditions to national government oppression, and seemingly nonviolent civil unrest can snowball into deadly protests, as evidenced by the Arab Spring [Mallinson, 2014]. Social media allows people to voice their displeasure and discuss their individual motives for engaging in protest. Therefore, Twitter effectively aggregates protest related information and researchers use daily word counts, such as “protest”, “racism”, and “police”, to forecast civil unrest [Korkmaz et al., 2015].

Due to recent political instability and mass protests within Latin America, researchers are actively engaged in creating real-time civil unrest forecasting models to assist in the region [Korkmaz et al., 2015; Ramakrishnan et al., 2014; Chen and Neill, 2014]. Figure 3.3 displays the prevalence of daily protests in six Latin American countries from November 2012 to August 2014. On the country level, Argentina, Brazil, Colombia, Mexico, Paraguay, and Venezuela all experienced civil unrest at least 40% of the days throughout the two year period. Furthermore, in Brazil and Mexico there existed protests on more than 500 days of the approximately 600-day period.

The current performance baseline of forecasting civil unrest at the country level in Latin America produces F1 scores in the range of 68% to 95% [Korkmaz et al., 2015].

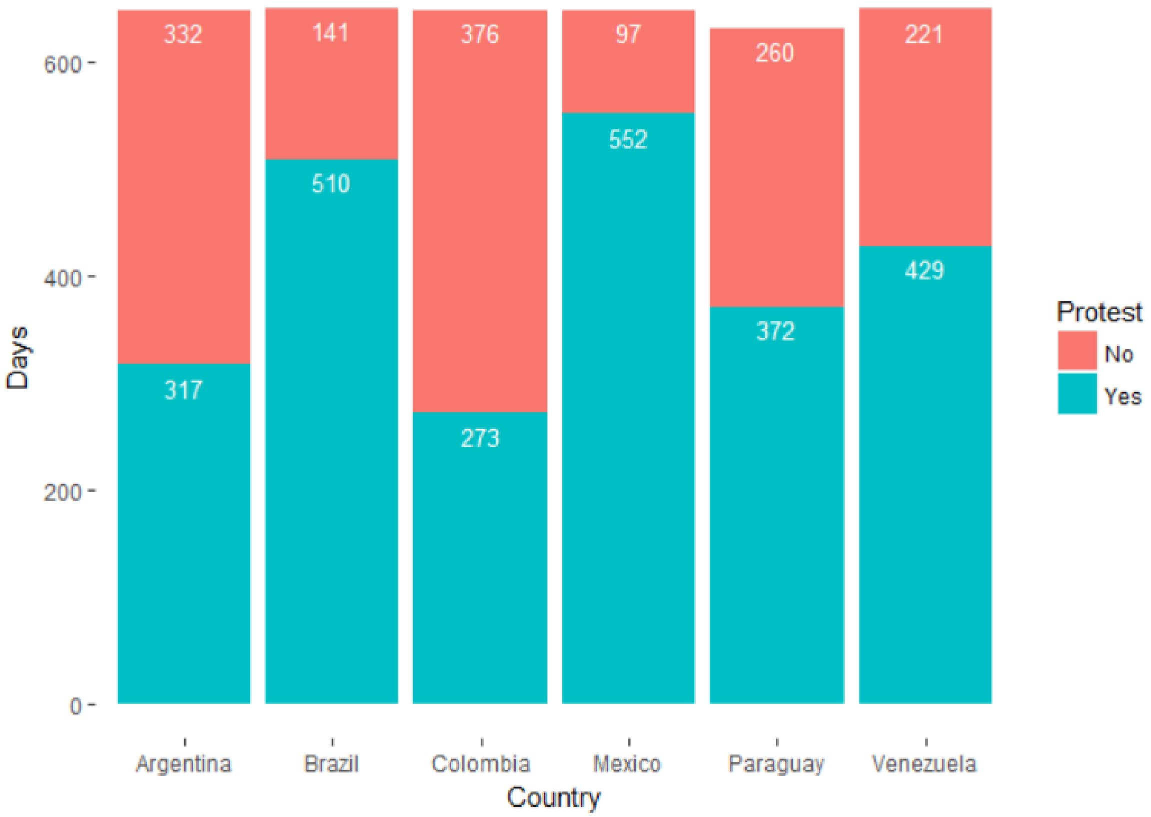


Figure 3.3: The number of protest and non-protest days for Argentina, Brazil, Colombia, Mexico, Paraguay, and Venezuela from November 2012 to August 2014 according the the Gold Standard Report.

The authors use logistic regression to forecast the probability of civil unrest in the six countries of Figure 3.3 from November 2012 to August 2014 and LASSO regularization to find a parsimonious model. The regularization allows the researchers to find predictive features in the high dimensional modeling scenario $(p > n)$ and infer the reasons for which people protest. That is, if the term "police" is predictive of civil unrest, this may require a different allocation of resources than if the term "unemployed" is predictive. Ground truth of the binary outcome, daily civil unrest, is produced by social scientists within the region and is reported in the Gold Standard Report (GSR). The authors use daily word counts from Twitter related to civil unrest and other features including blogs, news, currency exchange rates and Tor (essentially anonymous Twitter) daily access log counts. The dictionary of 962 protest-related Twitter terms for each country was created by subject matter experts on Latin America and contains words like “revolution” and phrases like “walk for peace”. Figure 3.4 displays a word cloud of the dictionary.

Figure 3.4: Word cloud of the 962 term dictionary of protest related words, phrases, and political leaders used to filter tweets and collect daily word counts. Terms are randomly scaled for the purpose of this graphic only.

3.8.2 Results

We now compare the logistic regression with LASSO regularization baseline method to the dynamic logistic regression and variable selection approach presented in Section 3.6. We model civil unrest in Argentina, Brazil, Colombia, Mexico, Paraguay, and Venezuela using data collected by Korkmaz et al. [2015]. We use the GSR as ground truth and include only Twitter terms as model features, $p = 962$. For more efficient implementation via parallelization, we model the first 600 observations in 50 day increments. In each 50 day period, $(1-50, 51-100, \ldots, 551-600)$, we fit each model on the first 25 observations and then forecast one day and five days ahead. We then move one day ahead in time, fit the model to the past data, and forecast again. The process is repeated until the last day is reached. For each of the six countries we use 1000 MCMC iterations to approximate the posterior distribution after a 100 iteration burn-in period. Forecast results are displayed in Table 3.7 and the top four most commonly selected variables are arranged in Table 3.8.

The recall of each model is markedly similar to the simulation results discussed in Section 3.7.3, with ranges of 97% to 100% for each model and country. Both the static and dynamic models rarely forecast no civil unrest when there is in fact a future protest looming. The precision for the dynamic model is also high for each country with ranges of 98% to 100%, indicating few false prediction of future protest. For the LASSO model however, the precision varies wildly between countries. For example, the precision for Brazil and Mexico are both in the lower to mid 80% range and the precision for Argentina is in the mid 50% range. As discussed in the simulation results of Section 3.7.3, the LASSO model tends to over forecast the probability of an event and therefore, the inconsistency of the precision for the civil unrest application can best be explained by Figure 3.3. Brazil and Mexico have the highest proportion of protests at 0.78 and 0.85 respectively. Countries where the proportion of protests is closer to 0.50, such as Argentina and Colombia, the precision for the LASSO model is closer to 50%.

Averaging the precision and recall results for the dynamic model, the F1 scores remain in the range 97% to 100%, evidencing a highly accurate civil unrest forecasting model. In both Argentina and Venezuela, the one day head forecasts were perfect over the entire 600 day period. In all cases, moving from a 1 day ahead forecast to five days ahead reduced the forecast accuracy as expected, indicating there is in fact a dynamic component to the unknown forces generating civil unrest. For the LASSO model, the F1 scores mimic the behavior of the precision, as the recall is invariably high. The F1 scores for Brazil and Mexico, the countries most likely to experience civil unrest, are in the 89% to 91% range. F1 scores for the other four countries are in the 60% to 81% range. Overall the dynamic model out performs the static model in forecasting civil unrest. The F1 scores for each country and forecast period are all higher for the dynamic logistic regression model than that of the baseline model.

Understanding the reasons people protest, and civil unrest inference in general, is achieved via variable selection. Table 3.8 reports the most common terms selected by each model over the two year forecasting period. For example, the most predictive terms for civil unrest in Colombia according to the dynamic model are reform (reforma), judgment (sentencia), traditional (tradicional), and environment (ambiental). In addition, the dynamic model selected these variables for model inclusion 25%, 23%, 23%, and 19% of the time respectively. Thus, for approximately 6 months of the two year time period, Colombians were presumably protesting reform of traditional values and environmental impact. In fact, in 2013 the Colombian government deployed $50,000$ troops to stymie the violent protests rooted in its citizens demanding reform of the Colombian agricultural business and unfair trade agreements forcing farmers out of business [CNN, 2013]. The

Table 3.7: Civil unrest forecast results for each country, model, and forecast period.

Country	Model	Forecast Days	Precision	Recall	F1 Score
Argentina 	LASSO	1	54.06	98.16	69.72
		5	55.18	97.85	70.56
	DGLM	1	100	100	100
		5	98.64	98.26	98.45
Brazil 	LASSO	1	80.33	100	89.09
		5	80.43	99.70	89.04
	DGLM	1	99.17	99.59	99.38
		5	99.32	99.51	99.41
Colombia 	LASSO	1	43.10	100	60.24
		5	43.04	99.44	60.08
	DGLM	1	100	98.43	99.21
		5	99.78	97.77	98.77
Mexico 	LASSO	1	83.85	99.59	91.04
		5	84.29	99.15	91.11
	DGLM	1	99.46	99.94	99.70
		5	98.99	99.90	99.45
Paraguay 	LASSO	1	53.69	98.76	69.56
		5	52.17	98.64	68.24
	DGLM	1	99.72	99.65	99.69
		5	98.31	98.24	98.27
Venezuela 	LASSO	1	67.33	100	80.47
		5	64.76	99.63	78.49
	DGLM	1	100	100	100
		5	98.27	97.42	97.85

protests revived in 2014 when the Colombian government failed to keep promises which quelled the riots of 2013 [BBC, 2014].

Most of the predictive terms reported in Table 3.8 provide some degree of inference regarding the reason for protest. For example, in Mexico, a highly traditional and religious country, the term homophobia (homophobia) is the third most commonly selected term according to the dynamic model, which indicates groups protesting for and against gay rights. On the other hand, in Argentina, the dynamic model finds terms such as warning (advertencia) and assembly (asamblea) to be predictive of civil unrest. These terms do not necessarily provide context to the protest, rather they are simply words that must be monitored within the Twitter feed to forecast the likelihood of future protests.

Table 3.8: The top four most commonly selected variables for each country and model.

Country	Model	Twitter Features (% of Time Selected)
Argentina	LASSO	Seguridad(9) Iniciativa(8) Efectivo(7) Proyecto(7)
	DGLM	Salud(22) Advertencia(21) Asamblea(19) Prohibir(18)
Brazil	LASSO	Atrocidad(14) Incremento(8) Laa(8) Legalizacia(7)
	DGLM	Accidente(18) Arara(17) Ministro(16) Masacre(16)
Colombia	LASSO	Effectivo(10) Electricidad(9) Justificacia(8) Directivos(8)
	DGLM	Reforma(25) Sentencia(23) Tradicional(23) Ambiental(19)
Mexico	LASSO	Rumores(9) Humillar(8) Plantear(7) Dictadura(7)
	DGLM	Excesos(17) Marcha(17) Homofobia(16) Caro(16)
Paraguay	LASSO	Contaminar(8) Procesados(8) Derechos(7) Afectados(7)
	DGLM	Embargo(25) Encontrar(25) Huelga(23) Destrozar(22)
Venezuela	LASSO	Prejuicios(10) Aumento(9) Violar(8) Perseguir(7)
	DGLM	Realizar(25) Hidroela(21) Derivar(17) Recuerdos(17)

As expected, the most predictive terms between countries are different given a specific model. Civil unrest occurs for a multitude of reasons and one nation's predictive terms, or motives for protest, do not correlate or influence another. The variable selection results also show the most commonly selected terms are completely different for each model within a given country. This is counterintuitive given the variable selection simulation results of Section 3.7.3. Under controlled conditions, the two models performed relatively similar in terms of the true positive and true negative rates. For the civil unrest application, the four most commonly selected variables for the dynamic model are selected between 16% and 25% of the time and for the LASSO model the range is reduced to only 6% to 14%. In this setting the LASSO model appears to select different terms for each 50 observation period. Conversely, the dynamic model identifies terms which are predictive over several months.

3.9 Conclusion

In this chapter we have presented combined model fitting and variable selection methodology for dynamic logistic regression. We include the Pólya-Gamma latent variable into the joint posterior distribution to more efficiently sample draws of state vectors using the Forward Filtering Backward Sampling algorithm. After model fitting, we use the estimated state vector at time t and its covariance to create penalized credible regions for variable selection. This method provides an entire solution path for the modeler to select the best of only p possible models. Furthermore, one can do variable selection dynamically using joint credible regions, or simply, at each time point a new observation becomes available and the state vector is updated.

The proposed methodology was applied to the problem of civil unrest in Latin America. We forecast the probability of future protest one day and five days ahead in the countries of Argentina, Brazil, Colombia, Mexico, Paraguay, and Venezuela. Using only protest-related terms as model features extracted from Twitter, we show improved accuracy compared to the baseline static logistic regression with LASSO regularization model. The F1 scores enhanced from a range of $60\% - 92\%$ for the baseline to $97\% - 100\%$ under dynamic logistic regression. The dynamic model is able to forecast according to the most recent information and account for dependencies between successive observations. Furthermore, the flexibility of the model captures the inherent dynamic nature of Twitter and allows the protest predictive terms to vary in time. The proposed variable selection technique dynamically selects predictors and captures the fluid reasons for civil unrest.

Chapter 4

Dynamic Logistic Regression Measurement Error

4.1 Introduction

Methodology to forecast events using tweets with imprecise locations has yet to be explored in the Twitter literature. In some cases, geographic clarity is not required, such as predicting stock price movement [Bollen et al., 2011] or box office revenue for the film industry [Asur and Huberman, 2010]. Conversely, geographic specific models, such as forecasting civil unrest in Latin America [Korkmaz et al., 2015] or influenza-like illness (ILI) rates in the United States [Achrekar et al., 2012; Culotta, 2010; Li and Cardie, 2013], require location-specific data. Thus far, geographic-specific models have been formed using only tweets containing their origins.

When using word counts to forecast civil unrest or influenza, the granularity of predictions are at the country level. A more informative model, which could be used to better allocate resources, would forecast these events at much finer geographic regions such as state, city, or meaningfully defined regions. Predicting at finer granularities of region size comes at a cost. The size of the available training data is related to the size of the prediction region. The smaller the region size, the smaller the data set available. Since only approximately 1% of tweets contain their origins [Ajao et al., 2015], it is necessary to use tweets with imputed geotags to create a more geographically narrow model.

Using only the best point estimate from an imputed geotag results in either including or excluding the tweet in subsequent analyses based on its location. As the region size decreases, the number of imputed geotags in the region inevitably reduces as well, leading to an eventual data sparsity problem. The hybrid geotag imputation model of Chapter 2 accounts for uncertainty in the location estimate, providing a probability measure of how likely the tweet originated from the region of interest. By accounting for this uncertainty in the forecast model, a data set with few geographically specific tweets will still maintain some forecasting information.

In this chapter we combine the geotag imputation approach of Chapter 2 and the dynamic logistic regression model of Chapter 3. Using word counts as model features effectively passes uncertainty from the imputed geotag to the secondary analysis. Our goal is twofold. We seek to understand how the uncertainty, or covariates measured with error, alters forecasting performance, and if accounting for uncertainty allows for the creation of geographic specific models at smaller region sizes. We compare two methods of estimating the periodic word counts, $\boldsymbol{X}_t$, on the performance of the dynamic logistic regression model.

1. Assume the imputed geotags are ground truth and $\boldsymbol{X}_t$ is fixed (the fixed covariate model, FCM).

2. Account for the uncertainty in the imputed geotags and let $\boldsymbol{X}_t$ be random (the

measurement error model, MEM).

This chapter is organized as follows. In Section 4.2 we first discuss how to transition the dynamic logistic regression model of Chapter 3, to one with measurement error in the covariates. We then discuss additional techniques to estimate its posterior distribution, such as Approximate Bayesian Computation, and detail the Gibbs sampler to fit the new model. In Section 4.3 we conduct a simulation to compare forecast performance between dynamic logistic regression models assuming the covariates are fixed and accounting for possible error in the covariates. Finally, we apply our methodology to monitor region specific influenza rates within the United States in Section 4.4 and end with a discussion in Section 4.5.

4.2 Model

In this section we transition the dynamic logistic regression model of Chapter 3 with known covariates to one with Berkson measurement error. We then detail the expanded joint posterior distribution and additional simulation techniques used to draw posterior samples.

4.2.1 The Dynamic Logistic Regression Model

Recall from Chapter 3, the *observation equation* for the dynamic logistic regression model is specified as

$$y_t|\boldsymbol{\beta}_t \sim Bernoulli(\pi_t) \ , \quad \pi_t = \frac{e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}}{1+e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}} \quad t = 1, \ldots, T \ , \tag{4.1}$$

where the response $y_t \in \{1, 0\}$ represents a "success" or "failure" respectively at time

t, $\boldsymbol{\beta}_t$ is the p-dimensional state vector, and $\boldsymbol{X}_t = (X_{t1}, \ldots, X_{tp})$ is the vector of known covariates. Also, the *state equation* is

$$\boldsymbol{\beta}_t = \boldsymbol{G}_t \boldsymbol{\beta}_{t-1} + \boldsymbol{w}_t \ , \qquad \boldsymbol{w}_t \sim N(\boldsymbol{0}, \boldsymbol{W}) \ , \tag{4.2}$$

where $\boldsymbol{G}_t$ is assumed to be a $p \times p$ identity matrix, and $\boldsymbol{w}_t$ is an independent sequence of independent Gaussian errors with mean zero and known variance components.

In the case of Twitter data, $\boldsymbol{X}_t$ is the vector of day t word counts collected for a region specific model. The covariates are calculated as

$$\boldsymbol{X}_t = \boldsymbol{O}_t + \boldsymbol{e}_t \ , \tag{4.3}$$

where $\boldsymbol{O}_t = (O_{t1}, \ldots, O_{tp})$ and $\boldsymbol{e}_t = (e_{t1}, \ldots, e_{tp})$ are the vectors of daily word counts for geotagged and imputed geotagged tweets respectively. Typically, $\boldsymbol{e}_t$ is assumed fixed so that imputed geotags are considered ground truth and uncertainty in the imputed location is ignored. Chapter 3 details the model fitting techniques for dynamic data when the covariates are fixed.

4.2.2 Berkson Measurement Error

In Chapter 2 we created the hybrid geotag imputation model, $h(y|m, n_j)$, using Gaussian mixture models to combine unigrams in the Twitter message, m, and user networks, n_j. Recall, the hybrid model estimates the probability of each point, y, within a spatial domain of being the origin of the tweet. Since the estimate of the origin is a probability distribution, the hybrid model provides a natural measure of uncertainty in the tweet originating from a given region. The estimated probability a tweet originated from region $\boldsymbol{R}$, is simply the cumulative density of the hybrid model within the region, $\int_{\boldsymbol{R}} h(y|m, n_j)$.

These probabilities can be used to better estimate the true daily word counts $\boldsymbol{X}_t$.

In order to account for the uncertainty in imputed geotags, and therefore the measurement error in daily word counts, we transition Equation (4.3) to an additive Berkson error model [Carroll et al., 2006]. Considering Berkson measurement error structure, the unobserved truth is a function of observed data and problem specific error.

- $\boldsymbol{X}_t$ are the "unobserved" true daily word counts within region $\boldsymbol{R}$.
- $\boldsymbol{O}_t$ are the "observed" daily word counts from tweets containing an origin in region $\boldsymbol{R}$.
- $\boldsymbol{e}_t$ are the "estimated" daily word counts in region $\boldsymbol{R}$ from tweets with imputed geotags.

Typically, in Berkson measurement error models, the errors are normally distributed with mean zero, $e_{ti} \sim N(0, \sigma^2)$. This results in the observed data being an unbiased estimate of the true covariates, $E[\boldsymbol{X}_t] = \boldsymbol{O}_t$, and therefore, model parameters are unbiased as well [Carroll et al., 2006].

For the application at hand, let the estimated daily word counts follow a Poisson-Binomial distribution,

$$e_{ti} \sim PoiBin\left(\boldsymbol{P} = \left[p_1 = \int_R h_1(y|m, n_j), \ldots, p_n = \int_R h_n(y|m, n_j)\right]\right).$$

The Poisson-Binomial distribution is a discrete probability distribution for the sum of independent Bernoulli trials that are not identically distributed and with support $\{0, 1, \ldots, n\}$ [Hong, 2013]. Here n represents the number of individual tweets collected on day t containing word i. This distribution allows us to estimate the daily word counts from imputed geotags and use different probabilities, $\boldsymbol{P} = \{p_1, \ldots, p_n\}$, for the estimated density within the region of interest for each tweet. Since observations are not identically distributed, the mean and variance of the Poisson-Binomial distribution are calculated as

$$E[e_{ti}] = \mu_{ti} = \sum_{j=1}^{n} p_j, \qquad VAR[e_{ti}] = \sigma_{ti}^2 = \sum_{j=1}^{n} p_j(1-p_j) \ .$$

As a result, the expected value of word counts for day t and unigram i is

$$E[X_{ti}] = E[O_{ti}] + E[e_{ti}] = O_{ti} + \sum_{j=1}^{n} p_j \ ,$$

where $\sum_{j=1}^{n} p_j$ represents the total density of n tweets within region R for the given day and unigram.

The Poisson-Binomial distribution is used in a variety of applications when practitioners seek to estimate the sum of n independent and non-identically distributed Bernoulli trails. For example, in reliability Hong et al. [2009] predict the total number of failures for a fleet of high-voltage power transformers where individual units have different failure probabilities, and in econometrics Duffie et al. [2007] predict the number of corporate defaults according to each company's unique balance sheet. Other examples can be found in engineering [Fernández and Williams, 2010], survey sampling [Chen and Liu, 1997], actuarial science [Pitacco, 2007], and bioinformatics [Niida et al., 2012].

In these applications, considerable detail is given to computation of the Poisson-Binomial probability mass function and cumulative distribution function. Given the support, $\{0, 1, \ldots, n\}$, the PMF and CDF are calculated as

$$P(e_{ti} = k) = \sum_{A \in B_k} \left(\prod_{j \in A} p_j \prod_{j \in A^c} (1-p_j) \right) , \qquad F_{e_{ti}}(k) = \sum_{m=0}^{k} P(e_{ti} = m) \ ,$$

where B_k is all subsets for which k integers can be selected. In order to compute $F_{e_{ti}}(k)$, we must enumerate all subsets in B_k, which of course is not practical. More efficient methods for calculating the distribution have been proposed. The discrete Fourier transformation of the Poisson-Binomial characteristic function provides a better estimate than the brute force method above, but is still slow for large sample sizes. Recursive calculation starting from $P(e_{ti} = 0)$ and calculating the successive probabilities is also computationally demanding. As a result, both the Poisson and Normal approximations can be used with estimates of the mean and variance calculated above. However, these two approximations perform poorly when n is small. The refined Normal approximation adjusts the skewness of the Poisson-Binomial distribution in order to correct for small sample sizes. The CDF is calculated as

$$F_{e_{ti}}(k) \approx G\left(\frac{k + 0.5 - \mu_{ti}}{\sigma_{ti}}\right), \qquad G(x) = \Phi(x) + \gamma_{ti}(1 - x^2)\phi(x)/6\ ,$$

where $\Phi(\cdot)$ is the standard normal CDF, $\phi(\cdot)$ is the standard normal PDF and

$$\gamma_{ti} = VAR[e_{ti}]^{-3/2}E[e_{ti} - \mu_{ti}]^3 = \sigma_{ti}^{-3}\sum_{j=1}^{n} p_j(1 - p_j)(1 - 2p_j)\ .$$

The refined NA makes a correction to the skewness of the Poisson-Binomial distribution. All techniques described above are available in the *PoiBin* R package [Hong, 2013].

4.2.3 Fitting the Dynamic Model with Measurement Error

To account for the uncertainty in daily word counts, we now assume the covariates are Poisson-Binomial distributed,

$$\pi(\boldsymbol{X}_t|\boldsymbol{O}_t, \boldsymbol{\Theta}) \sim PoiBin(\boldsymbol{P}) \ . \tag{4.4}$$

Recall the vector of probabilities, $\boldsymbol{P}$, is a function of the hybrid density estimate, $h(y|m, n_j)$. In order to estimate $\boldsymbol{P}$ we need to fit the hybrid geotagging model for the corpus of tweets and estimate the parameters of the Gaussian mixture models for ungirams and networks,

$$\boldsymbol{\Theta} = \{\theta_{j1}, ..., \theta_{jc_j}, \mu_{j1}, ..., \mu_{jc_j}, \Sigma_{j1}, ..., \Sigma_{jc_j}, \phi_{i1}, ..., \phi_{ic_i}, \mu_{i1}, ..., \mu_{ic_i}, \Sigma_{i1}, ..., \Sigma_{ic_i}\} \ . \tag{4.5}$$

The distribution of $\boldsymbol{X}_t$ depends on the fixed observed daily word counts, $\boldsymbol{O}_t$, and the Poisson-Binomial distribution parameters, $\boldsymbol{P}$, which are a function of the hybrid geotagging model parameters, $\boldsymbol{\Theta}$.

The joint posterior density for the dynamic logistic regression model with measurement error is written as

$$\pi(\boldsymbol{y}_{1:T}, \boldsymbol{\beta}_{1:T}, \boldsymbol{W}, \boldsymbol{\omega}_{1:T}, \boldsymbol{X}_{1:T}, \boldsymbol{\Theta}) = \tag{4.6}$$

$$\pi(\boldsymbol{\beta}_0) \times \pi(\boldsymbol{W}) \times \pi(\boldsymbol{\Theta}) \times$$

$$\prod_{t=1}^{T} \pi(y_t|\boldsymbol{X}_t, \boldsymbol{\beta}_t) \times \prod_{t=1}^{T} \pi(\boldsymbol{\beta}_t|\boldsymbol{\beta}_{t-1}, \boldsymbol{W}) \times \prod_{t=1}^{T} \pi(\omega_t|\boldsymbol{X}_t, \boldsymbol{\beta}_t) \times \prod_{t=1}^{T} \pi(\boldsymbol{X}_t|\boldsymbol{O}_t, \boldsymbol{\Theta}) \ ,$$

where the individual components are as follows.

- $\pi(y_t|\boldsymbol{X}_t, \boldsymbol{\beta}_t)$ is the observation equation from the exponential family.

- $\pi(\boldsymbol{\beta}_t|\boldsymbol{\beta}_{t-1}, \boldsymbol{W})$ is the normally distributed state equation.
- $\pi(\boldsymbol{X}_t|\boldsymbol{O}_t, \boldsymbol{\Theta})$ are the Poisson-Binomial distributed covariates.
- $\pi(\omega_t|\boldsymbol{X}_t, \boldsymbol{\beta}_t)$ is the Pólya-Gamma distributed latent variable.

Recall, the Pólya-Gamma latent variable eases the burden of simulation by providing pseudo-Gaussian observations. This allows for the use of the forward filtering backward sampling algorithm to sample from the states, $\boldsymbol{\beta}_{1:T}$. The individual priors are as follows.

- Let $\pi(\boldsymbol{\beta}_0)$ be the normally distributed conjugate prior for the initial state vector which initializes the Markov chain. The posterior for $\boldsymbol{\beta}_t$ is then used as the prior for $\boldsymbol{\beta}_{t+1}$
- Let $\pi(\boldsymbol{W})$ be the prior for the state equation covariance structure. Assume the covariance structure is $\boldsymbol{W} = diag\left(\frac{1}{\tau_1}, \ldots, \frac{1}{\tau_p}\right)$. Therefore, the prior can be written as a product due to its diagonal form, $\prod_{i=1}^{p} \pi(\tau_i)$. Let the inverse variance components be gamma distributed, $\pi(\tau_i) \sim gamma(\alpha, \gamma)$, to provide a conjugate prior.
- Let $\pi(\boldsymbol{\Theta})$ be the prior for each hybrid model parameter from Equation 4.5. Assuming each parameter is indepednent of one another, rewrite $\pi(\boldsymbol{\Theta})$ as a product of priors, $\prod_{k=1}^{|\boldsymbol{\Theta}|} \pi(\boldsymbol{\Theta}_k)$, where $\boldsymbol{\Theta}_k$ denotes the kth element of the set. Let the prior for each parameter be $\pi(\boldsymbol{\Theta}_k) \sim Unif(\alpha_k, \gamma_k)$.

Given that the posterior density is completely specified, now draw posterior samples of $\{\boldsymbol{\beta}_{1:T}, \boldsymbol{W}, \boldsymbol{\omega}_{1:T}, \boldsymbol{X}_{1:T}, \boldsymbol{\Theta}\}$ from the joint distribution in Equation (4.6). The states, variance components, and latent variable can be drawn with the same techniques as Chapter 3 since the full conditional distributions assume the covariates are fixed.

4.2.4 Approximate Bayesian Computation

The addition of the hybrid model parameters into the dynamic model causes the likelihood to become intractable. The individual probabilities for the Poisson-Binomial distribution are replaced with Gaussian mixture models and the dimensionality of $\boldsymbol{\Theta}$ results in a computational burden. Furthermore, the curse of dimensionality is present as the cardinality of $\boldsymbol{\Theta}$ grows as the size of the Twitter corpus grows. In order to obtain posterior draws from the dynamic logistic regression model with measurement error proposed in Section 4.2.3, we rely on Approximate Bayesian Computation (ABC).

As its name implies, Approximate Bayesian Computation is a method for approximating posterior distributions. Computing the likelihood in modern problems, such as the application at hand, is not always possible or computationally feasible. Approximate Bayesian Computation allows practitioners to enjoy the benefits of Bayesian analysis while also sidestepping the likelihood [Sunnåker et al., 2013]. Originally developed in the field of population genetics [Pritchard et al., 1999], ABC has expanded to numerous other scientific disciplines. Within the last decade, ABC has been utilized in epidemiology [Tanaka et al., 2006], inference for stereological extremes [Bortot et al., 2007], dynamical systems [Toni et al., 2009], and several evolutionary applications [Fearnhead and Prangle, 2010].

To bypass evaluation of the likelihood, the innovation in ABC is to simulate a draw of model parameters from the prior distribution and then consider it a posterior draw if it produces sample data "similar" to the observed data. To illustrate the full technique in the general case, let $\boldsymbol{D}$ be observed data, $\boldsymbol{\theta}$ be model parameters, $\boldsymbol{S}$ be computed summary statistics, $\rho(\cdot)$ be a distance metric, and ϵ be user specified tolerance. The original ABC-Rejection algorithm proceeds as follows [Grelaud et al., 2009].

1. Simulate $\boldsymbol{\theta}'$ from the prior distribution $\pi(\boldsymbol{\theta})$.

2. Simulate $\boldsymbol{D}'$ from the likelihood distribution $\pi(\boldsymbol{D}|\boldsymbol{\theta}')$ and compute $\boldsymbol{S}'|\boldsymbol{D}'$.

3. If $\rho(\boldsymbol{S}, \boldsymbol{S}') \leqslant \epsilon$ accept $\boldsymbol{\theta}'$ as a posterior draw.

4. Repeat until N samples of $\boldsymbol{\theta}'$ are drawn.

The performance of Approximate Bayesian Computation is a function of the prior and tolerance. If the tolerance is suitably small, then $\pi(\boldsymbol{\theta}|\rho(\boldsymbol{S}, \boldsymbol{S}') \leqslant \epsilon)$ is a good approximation to the target posterior distribution [Sunnåker et al., 2013]. Setting the tolerance too small comes at a computational cost as more simulations are then required. Conversely, setting the tolerance too large will cause the posterior approximation to mimic the prior distribution [Beaumont et al., 2002].

To lessen the importance of the user specified prior, Markov Chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC) techniques have been extended to ABC. In ABC-MCMC, proposed draws of $\boldsymbol{\theta}'$ are conditioned on the previous accepted posterior draw. After reaching convergence, newly proposed values will be governed by the posterior distribution as opposed to the prior [Beaumont, 2010]. In ABC-SMC, the goal is to initialize a population of parameters, $\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(N)}$, and have them develop into a sample from the target distribution over N iterations [Sisson et al., 2007]. In each iteration, the parameters receive weights according to how well they fit the target distribution. The parameters are then sampled based on the weights in the next iteration. In either case, ABC-MCMC or ABC-SMC, the algorithm is more computationally expensive then the ABC-Rejection algorithm, especially when the tolerance is small.

For well defined priors, a large tolerance is inconsequential. In fact, for well defined priors, a large tolerance allows Approximate Bayesian Computation to fully explore the parameter space and efficiently draws posterior samples [Frazier et al., 2017]. Given the accuracy of the hybrid geotagging model detailed in Chapter 2, we consider confidence intervals for each parameter in $\boldsymbol{\Theta}$ to be well-defined priors. Thus, let the prior for each parameter be $\pi(\boldsymbol{\Theta}_k) \sim Unif(\alpha_k, \gamma_k)$, where α_k and γ_k are the lower and upper confidence intervals respectively, from the maximum likelihood estimates in Chapter 2. For speed, we use the ABC-Rejection algorithm with a large tolerance to sample posterior draws of $\boldsymbol{\Theta}$. For a single draw, the algorithm we adopt is as follows.

1. For $k = 1, \ldots, |\boldsymbol{\Theta}|$, simulate $\boldsymbol{\Theta}'_k$ from $\pi(\boldsymbol{\Theta}_k) = Unif(\alpha_k, \gamma_k)$.

2. Simulate 100 locations, $\boldsymbol{D}'$, from the appropriate tweet density $h(\boldsymbol{D}|m, n_j, \boldsymbol{\Theta})$ and compute $\boldsymbol{S}'|\boldsymbol{D}'$. Let $\boldsymbol{S}'$ be the mean squared error where each data point is compared to the proper cluster.

3. If $\rho(\boldsymbol{S}, \boldsymbol{S}') \leqslant \epsilon = 1000 \text{ km}^2$ accept $\boldsymbol{\Theta}'$ as a posterior draw. Let $\rho(\cdot)$ measure the difference between $\boldsymbol{S}$ and $\boldsymbol{S}'$.

4. Repeat until a $\boldsymbol{\Theta}'$ is accepted.

Typically ABC is deployed in cases where the likelihood is intractable for all parameters. However, some applications require the use of ABC for only a subset of parameters and common Bayesian simulation techniques can be applied for the others [Turner and Van Zandt, 2014]. For the dynamic measurement error model, the likelihood is only intractable for the hybrid geotagging model parameters, $\boldsymbol{\Theta}$. The dynamic parameters, $\{\boldsymbol{\beta}_{1:T}, \boldsymbol{W}\}$, can be simulated from their full posterior conditional distributions as detailed in Chapter 3. Thus, we apply ABC only to obtain posterior draws of $\boldsymbol{\Theta}$ and traditional Gibbs sampler techniques for the remaining parameters, $\{\boldsymbol{\beta}_{1:T}, \boldsymbol{W}, \boldsymbol{\omega}_{1:T}, \boldsymbol{X}_{1:T}\}$.

4.2.5 Gibbs Sampler

In Chapter 3, we detailed the dynamic logistic regression model fitting process when the covariates are assumed fixed. Here we detail the Gibbs sampler when accounting for measurement error in the covariates. Note, because the full conditionals of the parameters $\{\boldsymbol{\beta}_{1:T}, \boldsymbol{W}, \boldsymbol{\omega}_{1:T}\}$ assume the covariates are fixed, posterior draws are obtained equivalently to Chapter 3.

1. Fit Gaussian mixture models to the training unigrams and networks as detailed in Chapter 2.

2. Conduct $B = 100$ bootstrap samples of the unigram and network locations, $\boldsymbol{D}|u_i$ and $\boldsymbol{D}|n_j$ respectively, to find estimates of the 95% confidence intervals, (α_k, γ_k), for each parameter in $\boldsymbol{\Theta}$.

3. Initialize the latent variable vector $\boldsymbol{\omega}_{1:T}^{(0)}$, the states $\boldsymbol{\beta}_{1:T}^{(0)}$, and the state covariance $\boldsymbol{W}^{(0)}$.

4. For iterations $s = 1, \ldots, N$:

    (a) Sample Gaussian mixture model parameters $\boldsymbol{\Theta}^{(s)}$ using the ABC-Rejection algorithm.

    i. For $k = 1, \ldots, |\boldsymbol{\Theta}|$, simulate $\boldsymbol{\Theta}'_k$ from $\pi(\boldsymbol{\Theta}_k) = Unif(\alpha_k, \gamma_k)$, where α_k and γ_k are the lower and upper confidence intervals from the maximum likelihood estimates in Chapter 2.

    ii. Simulate 100 locations, $\boldsymbol{D}'$, from the appropriate tweet density $h(\boldsymbol{D}|m, n_j, \boldsymbol{\Theta})$ and compute $\boldsymbol{S}'|\boldsymbol{D}'$. Let $\boldsymbol{S}'$ be the mean squared error where each data point is compared to the proper cluster.

iii. If $\rho(\boldsymbol{S}, \boldsymbol{S}') \leqslant \epsilon = 1000\,\mathrm{km}^2$ accept $\boldsymbol{\Theta}'$ as a posterior draw. Let $\rho(\cdot)$ measure the difference between $\boldsymbol{S}$ and $\boldsymbol{S}'$.

iv. Repeat until a $\boldsymbol{\Theta}'$ is accepted and let $\boldsymbol{\Theta}^{(s)} = \boldsymbol{\Theta}'$.

(b) Sample a new set of covariates $\boldsymbol{X}_{1:T}^{(s)}$, from the full conditional distribution, which is a function of both the observation equation and the Poisson-Binomial covariate densities,

$$\pi(\boldsymbol{X}_{1:T}|\cdot) \;\propto\; \prod_{t=1}^{T} \pi(y_t|\boldsymbol{X}_t, \boldsymbol{\beta}_t) \times \prod_{t=1}^{T} \pi(\boldsymbol{X}_t|\boldsymbol{O}_t, \boldsymbol{\Theta})$$

$$\propto \prod_{t=1}^{T} \frac{exp(\boldsymbol{X}_t\boldsymbol{\beta}_t)^{y_t}}{1 + exp(\boldsymbol{X}_t\boldsymbol{\beta}_t)} \times \prod_{t=1}^{T}\prod_{i=1}^{p} \left[G\left(\frac{X_{ti} + 0.5 - \mu_{ti}}{\sigma_{ti}}\right) - G\left(\frac{X_{ti} - 0.5 - \mu_{ti}}{\sigma_{ti}}\right)\right]$$

For $t = 1, \ldots, T$ and $i = 1, \ldots, p$, sample each X_{ti} using the Metropolis-Hastings Algorithm.

i. Sample X_{ti}^{*} from the discrete proposal distribution,

$$J(X_{ti}|X_{ti}^{(s-1)}) \equiv DUnif\{X_{ti}^{(s-1)} - 2, X_{ti}^{(s-1)} - 1, X_{ti}^{(s-1)} + 1, X_{ti}^{(s-1)} + 2\}.$$

ii. Compute the acceptance ratio $r = \frac{\pi(X_{ti}^{*}|\cdot)}{\pi(X_{ti}^{(s-1)}|\cdot)}$.

iii. Let $X_{ti}^{(s)} = X_{ti}^{*}$ with probability $min(r, 1)$ and set $X_{ti}^{(s)} = X_{ti}^{(s-1)}$ otherwise.

(c) Sample $\boldsymbol{\beta}_{1:T}^{(s)}$ using the FFBS algorithm.

(d) Sample the components of $\boldsymbol{W}^{(s)} = diag\left(\frac{1}{\tau_1^{(s)}}, \ldots, \frac{1}{\tau_p^{(s)}}\right)$ individually from the updated conjugate prior,

$$\pi\Big(\tau_i^{(s)}|\cdot\Big) \sim Gamma\Bigg(\alpha + \frac{T}{2}, \gamma + \sum_{t=1}^{T}(\beta_{ti}^{(s)} - \beta_{(t-1)i}^{(s)})^2\Bigg) \, .$$

(e) Sample each element of the latent variable vector $\omega_t^{(s)}$, from the Pólya-Gamma distribution conditioned on the covariates and states, $PG(1, \boldsymbol{X}_t^{(s)}\boldsymbol{\beta}_t^{(s)})$, for $t = 1, \ldots, T$.

4.3 Simulation

For this simulation we will mimic a data collection scenario from the Twitter API with a few assumptions and restrictions in the data generating process for tractability. The goal of this simulation is to compare the forecast performance of the dynamic logistic regression models with assumed fixed and random covariates. We provide conditions for the fixed covariate model to forecast with a similar sample size as the measurement error model. Approximately one fourth of all tweets generated will be within the region of interest and the number of tweets with observed geotags will be well above the real 1% Twitter API conditions. Although not realistic in practice, since the FCM will produce fewer tweets within the region of interest, this establishes if limitations exist in using the MEM.

4.3.1 Simulate Tweets

In the GMM construction of the tweets detailed below, notice that the centers of the clusters are simulated on a 10×10 grid with $(0, 0)$ as the center. The density of the tweets are naturally split into the four quadrants on the Cartesian plane. For this simulation, consider each quadrant a separate geographical region and consider our region of interest to be quadrant 1, $\boldsymbol{Q}_1$. We want to use daily word counts to predict a binary event in the region of interest. Furthermore, assume the unigrams we generate for this simulation, u_i for $i = 1, \ldots, 500$, are split into two sets. Let $\boldsymbol{U}_1 = \{u_1, \ldots, u_p\}$ be the unigrams we consider as possible predictors in our dynamic model, and let $\boldsymbol{U}_2 = \{u_{p+1}, \ldots, u_{500}\}$ be other unigrams used to generate tweets.

For each unigram u_i, we simulate a Gaussian mixture model representing the observed density of the unigram,

$$g(y|u_i) = \sum_{k=1}^{c_i} \phi_{ik} N(y|\mu_{ik}, \Sigma_{ik}) \; .$$

Here N is the bivariate normal density function with parameters μ_{ik} and Σ_{ik} as the mean vector and covariance matrix respectively. In addition, ϕ_{ik} are the mixture weights, the subscript k represents the individual mixture components for each GMM, and c_i is the number of mixture components. The parameters of $g(\cdot)$ are simulated as follows.

- We consider between 1 and 5 clusters of data, $c_i \sim DUnif\{1, \ldots, 5\}$.
- For each cluster we simulate a total number of locations, $l_k \sim DUnif\{10, \ldots, 100\}$. In turn we normalize the weights, $\phi_{ik} = \frac{l_k}{\sum_{k=1}^{c_i} l_k}$.
- For the two element vector of μ_{ik}, we simulate the x-axis mean and the y-axis mean as $\mu_x \sim Unif(-5, 5)$ and $\mu_y \sim Unif(-5, 5)$.
- For the 2×2 covariance matrix Σ_{ik}, the off diagonal elements are assumed zero, $\Sigma_{ik12} = \Sigma_{ik21} = 0$, and the variance components are simulated as $\Sigma_{ik11} \sim Unif(0.1, 3)$ and $\Sigma_{ik22} \sim Unif(0.1, 3)$.

- Simulate $\sum_{k=1}^{c_i} l_k$ observed locations for each unigram from the population, $\boldsymbol{D}|u_i \sim g(y|u_i)$.

Next, for each network n_j, for $j = 1, \ldots, 50$, we simulate a Gaussian mixture model representing the observed density of a user's definable network,

$$f(y|n_j) = \sum_{k=1}^{c_j} \theta_{jk} N(y|\mu_{jk}, \Sigma_{jk}) \ .$$

The number of components, c_j, as well as the parameters θ_{jk}, μ_{jk}, and Σ_{jk} are generated similarly as above.

- We consider between 1 and 2 clusters of data, $c_j \sim DUnif\{1, 2\}$.
- For each cluster we simulate a total number of locations, $l_k \sim DUnif\{10, \ldots, 100\}$. In turn, we normalize the weights, $\theta_{jk} = \frac{l_k}{\sum_{k=1}^{c_j} l_k}$.
- For the two element vector of μ_{jk}, we simulate the x-axis mean and the y-axis mean as $\mu_x \sim Unif(-5, 5)$ and $\mu_y \sim Unif(-5, 5)$.
- For the 2×2 covariance matrix Σ_{jk}, the off diagonal elements are assumed zero, $\Sigma_{jk12} = \Sigma_{jk21} = 0$, and the variance components are simulated as $\Sigma_{jk11} \sim Unif(0.1, 1)$ and $\Sigma_{jk22} \sim Unif(0.1, 1)$.
- Simulate $\sum_{k=1}^{c_j} l_k$ observed locations for each network from the population, $\boldsymbol{D}|n_j \sim f(y|n_j)$.

Notice the simulation of the number of clusters and the covariance components for those clusters differs from $g(\cdot)$ to $f(\cdot)$. Typically, there are fewer clusters defining a network GMM than a unigram GMM, and the network cluster tends to be smaller as well. Therefore, the network tends to be more predictive of location.

Next we simulate tweets. Recall, each individual tweet is a combination of multiple words and at most a single network. Therefore, to generate tweets we will randomly select unigrams and a network to join together and create a hybrid Gaussian mixture model, $h(\cdot)$.

- Let $T_m \sim DUnif\{2, \ldots, 5\}$ be the number of words selected for the tweet.
- Select T_m unigrams from the entire set to create the message, m. Select 1 unigram from $\boldsymbol{U}_1$ and $T_m - 1$ unigrams from $\boldsymbol{U}_2$. This restricts each tweet to have exactly 1 unigram considered for prediction and $T_m - 1$ additional noise unigrams.
- Select 1 network from the entire set to add a definable network to the message, $n_j \sim DUnif\{n_1, \ldots, n_{50}\}$.
- Combine the selected unigrams and network to create the hybrid density of the origin,

$$h(y|m, n_j) = \delta_0 f(y|n_j) + \sum_{l=1}^{T_m} \delta_l g(y|m_l) \ .$$

Compute the mixture weights, $\{\delta_0, \ldots, \delta_{T_m}\}$, as detailed in Chapter 2.

Given the geographically definable region of interest, $\boldsymbol{Q}_1$, simulate the quadrant of origin for each tweet.

- Let $z \in \{1, 0\}$ represent the origin of the tweet. $z = 1$ represents $\boldsymbol{Q}_1$ is the origin and $z = 0$ otherwise.
- Simulate z as a Bernoulli random variable,

$$z \sim Bernoulli\left(\iint_{\boldsymbol{Q}_1} h(y|m, n_j)\right) \ .$$

The origin of each tweet is stochastically related to the proportion of density in $\boldsymbol{Q}_1$.

To mimic Twitter data, we randomly censor 80% of the origin of tweets, z. Thus, 20% of tweets had an observable geotag from the Twitter API, and the other 80% require an imputed geotag.

- Let $v \in \{1, 0\}$ represent whether or not the origin of the tweet is censored. $v = 1$ indicates the origin is observed and $v = 0$ indicates it is censored.
- Simulate v as a Bernoulli random variable, $v \sim Bernoulli(0.20)$.

To create a corpus of tweets, simulate $s_t = p \times 40$ tweets for each day t. For notation, let the total number of tweets simulated on day t with unigram i be s_{ti}.

4.3.2 Simulate Dynamic Data

Given that the origin of tweets has been established, we can now define the vectors of true daily word counts and simulate dynamic logistic regression data. In each case of the simulation, the binary response is generated from the *observation equation*

$$y_t|\boldsymbol{\beta}_t \sim Bernoulli(\pi_t), \quad \pi_t = \frac{e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}}{1 + e^{\boldsymbol{X}_t\boldsymbol{\beta}_t}} \quad t = 1, \ldots, T \ ,$$

and the state vectors are generated from the *state equation*

$$\boldsymbol{\beta}_t = \boldsymbol{G}_t\boldsymbol{\beta}_{t-1} + \boldsymbol{w}_t \ , \qquad \boldsymbol{w}_t \sim N(\boldsymbol{0}, \boldsymbol{W}) \ .$$

The number of observations for each time series is varied in $T \in \{25, 50\}$ and the parameters freely vary according to a random walk by setting both the transition matrix and the state equation covariance matrix to the identity, $\boldsymbol{G}_t = \boldsymbol{W} = \boldsymbol{I}_p$. The number of candidate predictors is varied in $p \in \{25, 50, 75\}$ and the true number of predictors used to generate $\boldsymbol{y}_{1:T}$ is approximately ten percent of of p, $p^* \in \{3, 5, 8\}$. For each data set, p^* indices are randomly chosen as the nonzero locations of the state vectors $\boldsymbol{\beta}_t$. Finally, let the true daily word counts be the number of tweets originating from $\boldsymbol{Q}_1$ for each day t and unigram i, $\boldsymbol{X}_t = \{\sum_{b=1}^{s_{t1}} z_{t1b}, \ldots, \sum_{b=1}^{s_{tp}} z_{tpb}\}$.

4.3.3 Model Comparison

We seek to compare two methods of estimating the daily word counts, $\boldsymbol{X}_t$, on the performance of the dynamic logistic regression model.

1. Assume the imputed geotags are ground truth and $\boldsymbol{X}_t$ is fixed (the fixed covariate model, FCM).
2. Account for the uncertainty in the imputed geotags and let $\boldsymbol{X}_t$ be random (the measurement error model, MEM).

For scenario 1, assuming $\boldsymbol{X}_t$ is fixed, requires using the best point estimate from the hybrid density as ground truth for the imputed geotags of censored tweets. Therefore, the daily word counts are the total number of uncensored tweets and geotagged tweets in the region of interest. Mathematically, estimated daily word counts for scenario 1 is written,

$$X_{ti} = \sum_{b=1}^{s_{ti}} \left[\mathbb{1}\big(z_{tib} = 1\big)\mathbb{1}\big(v_{tib} = 1\big) \;\; + \;\; \mathbb{1}\big(v_{tib} = 0\big)\mathbb{1}\big(\arg\max_{y} \;\; h_{tib}(y|m, n_j) \;\; \in \;\; \boldsymbol{Q}_1\big)\right].$$

For scenario 2, accounting for the uncertainty in imputed geotags when fitting the model is discussed in Section 4.2.

4.3.4 Metrics

For each time series length, $T \in \{25, 50\}$, we fit the model on the first 50% of the observations, $\{13, 25\}$, and then forecast one time period ahead and five time periods ahead. We then move one unit of time ahead, fit the model to the past data and forecast again. The process is repeated until the end of the times series.

Performance is measured in the same way as Chapter 3. We use the confusion matrix to measure the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates using a cutoff of 0.50. We also report the recall $\left(\frac{TP}{TP+FN}\right)$, precision $\left(\frac{TP}{TP+FN}\right)$, and the harmonic average between the two metrics, the F1 scores.

4.3.5 Results

For each combination of time series length $T \in \{25, 50\}$ and candidate predictors $p \in \{25, 50, 75\}$, we average results across 50 data sets. For each data set we use 1000 MCMC iterations to approximate the posterior distribution after a 100 iteration burn-in period. Forecast results are displayed in Tables 4.1 and 4.2.

In each case of the simulation, the results are markedly similar comparing the fixed covariate model (FCM) and the measurement error model (MEM). For the forecast results, the FCM achieves a greater or equal true positive rate in all cases, and the models take turns in achieving a higher true negative rate. A similar pattern emerges for false predictions, with the lower percentage of false predictions varying between models.

The precision, recall, and F1 scores are all slightly higher for the FCM considering only the one time period ahead forecast results. The F1 scores range from $97\% - 100\%$ for the FCM and $94\% - 99\%$ for the MEM. The forecast performance of the MEM is more competitive with the FCM when considering the longer forecast period. Both models achieve F1 scores in the $88\% - 96\%$ range. Although the F1 scores are approximately equal in this scenario, the precision is higher for the FCM and the recall is higher for the MEM, indicating fewer false positive prediction committed by the FCM and fewer false negative predictions committed by the MEM.

As expected, the forecast results for each model generally improve when moving from a time series length of $T = 25$ to $T = 50$. Intuitively, the dynamic model has more time to estimate the model, leading to better performance. Also as expected, forecast performance for each model reduces as the number of candidate predictors increase from $p = 50$ to $p = 75$. However, the F1 scores remain above 91% for each model in the $p > T$ scenario.

Overall, the forecast results are highly similar between each model. Recall, the simulation conditions provided each model an adequate amount of data to forecast within the region of interest. Even under ideal conditions for the FCM, the MEM maintains comparable forecast performance under all scenarios. Accounting for uncertainty in the location estimates does not degrade performance.

The next section applies each model to the common task of forecasting influenza under more realistic conditions, where approximately 1% of word counts scraped from Twitter contain an origin and are collected globally. We demonstrate the effect of varying the region size on forecast performance. As the region size decreases and the amount of exploitable data reduces, the MEM performance relatively improves, as it uses the uncertainty in the imputed geotag to more accurately forecast the target.

Table 4.1: Forecast results for time series length $T = 25$.

Model	Forecast	Predictors	TP	TN	FP	FN	Precision	Recall	F1
FCM	1	25	40.00	58.00	1.25	0.75	96.97	98.16	97.56
	5	25	36.42	54.50	5.25	3.83	87.40	90.48	88.92
MEM	1	25	38.75	57.00	1.25	3.00	96.88	92.81	94.80
	5	25	36.08	54.92	4.83	4.17	88.19	89.64	88.91
FCM	1	50	52.25	47.25	0.25	0.25	99.52	99.52	99.52
	5	50	48.42	44.67	1.33	5.58	97.33	89.67	93.34
MEM	1	50	50.50	48.25	0.25	1.00	99.51	98.06	98.78
	5	50	47.67	45.33	1.33	5.67	97.29	89.37	93.16
FCM	1	75	37.00	61.75	0.50	0.75	98.67	98.01	98.34
	5	75	35.50	58.17	4.66	1.67	88.39	95.51	91.81
MEM	1	75	37.00	61.25	1.00	0.75	97.37	98.01	97.69
	5	75	34.67	59.17	3.66	2.50	90.45	93.27	91.84

Table 4.2: Forecast results for time series length $T = 50$.

Model	Forecast	Predictors	TP	TN	FP	FN	Precision	Recall	F1
FCM	1	25	35.00	64.20	0.40	0.40	98.87	98.87	98.87
	5	25	32.14	62.02	2.48	3.36	92.84	90.53	91.67
MEM	1	25	33.40	63.90	1.70	1.00	95.16	97.09	96.12
	5	25	31.48	62.38	3.02	3.12	91.25	90.98	91.11
FCM	1	50	58.00	41.60	0.20	0.20	99.66	99.66	99.66
	5	50	55.25	39.95	1.57	3.23	97.24	94.48	95.84
MEM	1	50	57.00	41.60	0.70	0.70	98.79	98.79	98.79
	5	50	54.05	41.19	2.33	2.43	95.87	95.69	95.78
FCM	1	75	52.50	46.25	0.75	0.50	98.59	99.06	98.82
	5	75	49.45	44.08	2.78	3.69	94.68	93.06	93.86
MEM	1	75	51.25	45.50	1.50	1.75	97.16	96.69	96.93
	5	75	48.05	45.79	3.15	3.01	93.85	94.10	93.98

4.4 Application

Recall from Chapter 2, we imputed geotags for influenza related tweets using the Eisenstein data set to monitor the health of the United States for one week in March of 2010. By stacking the spatial probability distributions of each imputed geotag, we were able to account for the uncertainty in the location estimate and accurately estimate the distribution of influenza using CDC reports as ground truth.

In this section, we revisit the application of influenza. Instead of estimating the most affected regions, we forecast the 2016-2017 influenza season in three regions of varying granularity: the contiguous United States, the South Atlantic region (Delaware, Florida, Georgia, Maryland, North Carolina, South Carolina, Virginia, West Virginia), and North Carolina. For each geographic specific model, we use weekly word counts of ten influenza related terms to forecast the prevalence of the disease: "cough", "doctor", "fever", "flu", "headache", "sick", "sneeze", "sore throat", "virus", and "vomit". Note, regular expression patterns of the candidate predictors are also included. For example, the words "coughs", "coughed", and "coughing" are all considered the same term, "cough". We compare the results of the dynamic logistic regression model for each region assuming the imputed geotag is fixed and when accounting for uncertainty in the estimate.

4.4.1 Data

Los Alamos National Laboratory houses a repository of tweets collected from the free Twitter streaming API. This equates to approximately 1% of all tweets generated globally. From this repository we collected tweets generated from the first week of October, 2016 to the last week of September, 2017. Specifically, we collected 102,688 tweets during this period containing at least one of the ten candidate influenza predictors. This sample includes 1,729 geotagged tweets and 100,959 tweets without an origin. Additionally, 42,629 random geotagged tweets were collected to increase the training data set size for geotag imputation. The 44,358 total geotagged tweets were then used to impute geotags using the hybrid model of Chapter 2. The origin of these tweets are plotted in Figure 4.1. Note, we do not restrict the collection of tweets to any particular region, as this would bias the geotag imputation model.

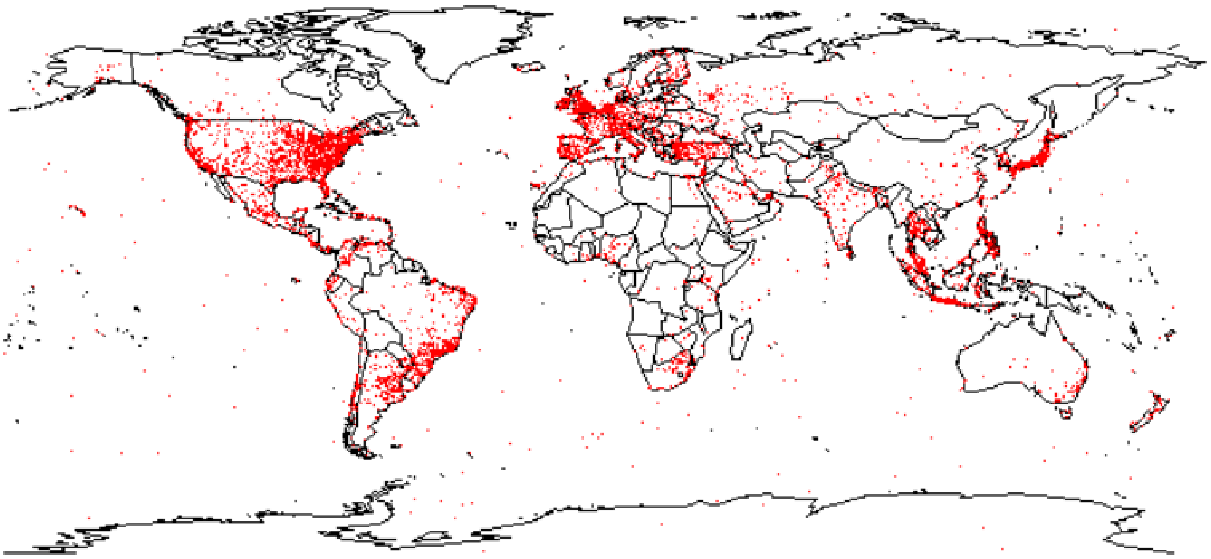


Figure 4.1: Locations of all $44,358$ tweets sampled to train the hybrid geotag imputation model.

Weekly influenza-like illness (ILI) rates for the national, regional, and state level are reported by the Centers for Disease Control and Prevention (CDC) [CDC, 2017]. Figure 4.2 displays the time series of weekly ILI rates for the 2016-2017 influenza season by each region considered for this analysis. Each region, demonstrates a general upward trend in the ILI rate from October, 2016 to approxiately March, 2017, when the rates decline until the following flu season. The maximum ILI rate is inversely related to the region size. North Carolina has the highest ILI rate, followed by the South Atlantic region and then the lower 48 states.

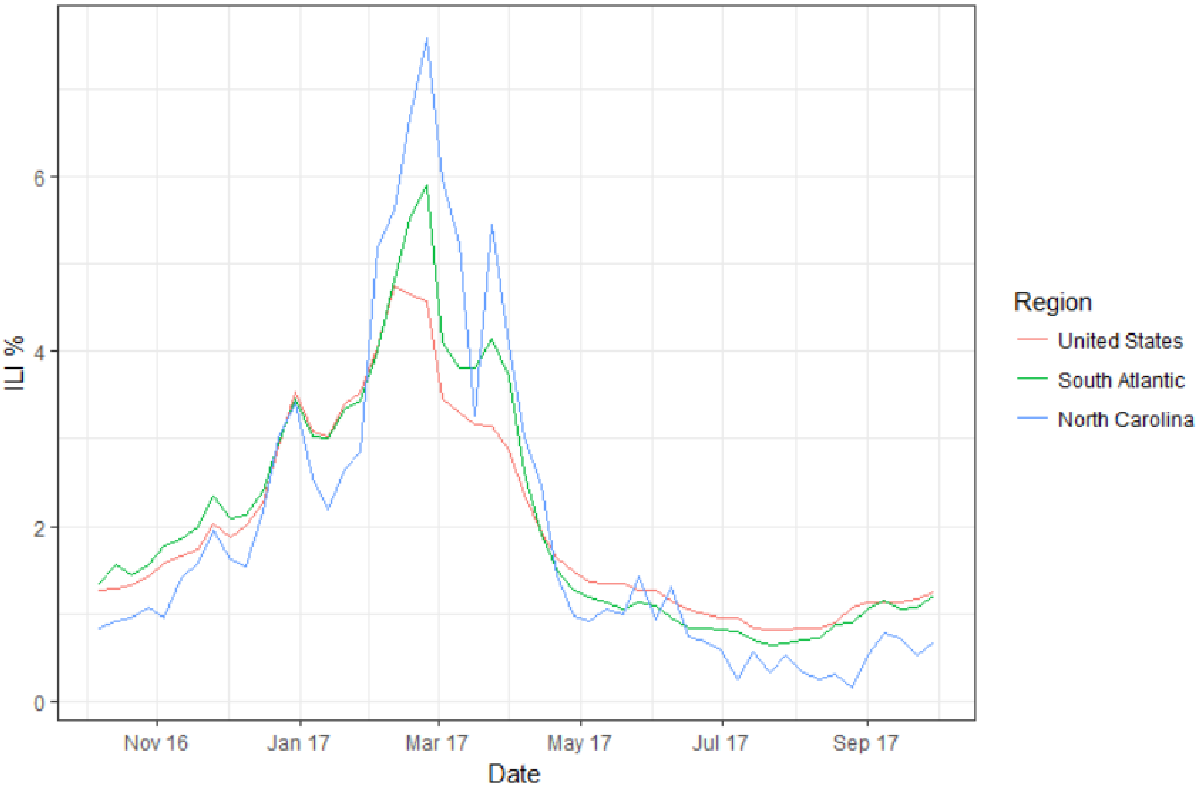


Figure 4.2: Weekly influenza-like illness (ILI) rates from October, 2016 to March, 2017 for the United States, South Atlantic, and North Carolina regions.

4.4.2 Results

To apply dynamic logistic regression, we use a 3% ILI cutoff rate for the response. That is, we forecast the probability that the weekly ILI rate is above 3% using influenza related weekly word counts scraped from Twitter. For the 52 week period, October, 2016 to September, 2017, we forecast one and five weeks ahead for each model starting at week 10. Finally, we use 1000 MCMC iteration to approximate the posterior distribution after a 100 iteration burn-in period. We now compare the geographic specific models, altering the assumption of fixed or random covariates. Forecast results are displayed in Table 4.3 and the frequency of influenza terms selected for model inclusion are arranged in Table 4.4.

Beginning with the largest region size, the forecast results are equivalent for both models given a forecast period. For the next week forecast, the precision, recall, and F1 scores are identical at 92.31%. The five week ahead forecast accuracy reduces to 76.92% for precision, and 80.64% for recall, lowering the F1 score to 78.74%.

Moving to the South Atlantic region, the FCM model outperformed the measurement error model for the one week ahead forecast due to perfect precision, indicating zero false positive predictions. However, the recall for each model was the same at 92.86%. For the five week ahead forecast the opposite result transpired. That is, the precision for the measurement error model is higher at 85.93% compared to 78.57% for the fixed covariate model and again, the recall is the same for each model at 82.09%. Due to varying precision, the F1 score is higher for the measurement error model, 83.97%, and lower for the fixed covariate model, 80.29%.

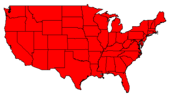
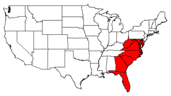
At the state level, the precision, recall, and F1 scores are all higher or equivalent for the measurement error model. For the one week ahead forecast, the measurement error model achieves perfect precision compared to 83.33% for the FCM. The recall is equivalent at 83.33%. The five week ahead forecast is significantly better for the MEM due to better precision, or fewer false positive predictions. The precision is 84.78% for MEM compared to only 61.67% for the FCM. This is the first scenario the recall differs between the two models with 70.91% for the MEM and 67.27% for the FCM. Aggregating the results, the F1 scores are 77.23% and 64.35% for the MEM and FCM respectively.

The forecast results show a significant drop in accuracy for the FCM at the smallest region. At the two larger region sizes the F1 scores range from $78\%-97\%$ and then reduce to a range of only $64\%-84\%$ at the state level. Conversely, the F1 scores are consistent for all region sizes when using the MEM, with F1 scores in the range of $77\% - 93\%$. This can best be explained by the amount of available data in each region. For the fixed covariate model, which assumes the imputed geotag is ground truth, only 44 tweets containing an influenza related term were geotagged to North Carolina. This is compared to 921 tweets for the South Atlantic region and $17,484$ tweets for the Nation. Thus, as the region size decreases, the amount of exploitable data also decreases, causing an adverse impact on model performance. By accounting for uncertainty in the imputed geotag, the measurement error model assumes there is some probability greater than zero that the tweet originated from the region, regardless of whether or not the best point estimate is within the region of interest. The total density of tweets within each region across all weeks, $\sum_{n=1}^{100,959} \int_R h_n(y|m, n_j)$, is approximately 34,921, 8,789, and 1,322 for the national, South Atlantic, and state regions respectively. Therefore, the measurement error model maintains enough information to forecast at the smallest region size, North Carolina.

The variable selection results in Table 4.4, conducted using the penalized credible region approach of Chapter 3, display further evidence of an inadequate sample size in the state region for the fixed covariate model. The FCM did not select a single variable for model inclusion in North Carolina. That is, the forecast results were based solely on a dynamic intercept model.

In general, regarding Table 4.4, the predictive terms selected for model inclusion tend to be dissimilar between each scenario of model and region size. For example, the term "virus" was included 62% of the time in the South Atlantic region for the FCM and

Table 4.3: Influenza forecast results for each region, model, and forecast period.

Country	Model	Forecast Weeks	Precision	Recall	F1 Score
United States	FCM	1	92.31	92.31	92.31
		5	76.92	80.64	78.74
	MEM	1	92.31	92.31	92.31
		5	76.92	80.64	78.74
South Atlantic	FCM	1	100	92.85	96.29
		5	78.57	82.09	80.29
	MEM	1	92.86	92.86	92.86
		5	85.93	82.09	83.97
North Carolina	FCM	1	83.33	83.33	83.33
		5	61.67	67.27	64.35
	MEM	1	100	83.33	90.91
		5	84.78	70.91	77.23

only 12% for the MEM. Likewise, the term “fever” was selected 19% of the time for the FCM in the same region and 64% of the time for the MEM. The only prominent similarity in variables selected for the two models is for the terms “flu” and “sick”. The term “flu” is the most selected term in the national region for both models, but is almost entirely excluded in the South Atlantic region. Instead, the term “sick” is most commonly selected in the South Atlantic region for each model. There exists further similarity in which variables are not selected, especially in the South Atlantic region. Both models agree that “flu”, “headache”, “sneeze”, and “sore throat” are not predictive of influenza at this level of geographic granularity.

As discussed previously, the FCM did not select a single variable for model inclusion in the state region. However, the opposite occurred for the MEM. All predictors except one, “sore throat”, were selected for model inclusion at least 50% of the time. This indicates a need for a diverse set of predictors as the region size reduces.

Table 4.4: The percentage of time each influenza term was selected for model inclusion by region.

Region	Model	Cough	Doctor	Fever	Flu	Headache	Sick	Sneeze	Sore Throat	Virus	Vomit
US	FCM	76	52	52	95	83	71	2	36	69	0
	MEM	24	86	29	93	5	7	31	19	93	93
SA	FCM	0	59	19	0	0	98	0	0	62	64
	MEM	14	2	64	2	2	95	0	0	12	2
NC	FCM	0	0	0	0	0	0	0	0	0	0
	MEM	98	71	64	50	100	98	98	12	62	98

4.5 Discussion

In this chapter we have presented methodology extending the dynamic logistic regression model to one with measurement error in covariates. The motivation for this work is to account for the location uncertainty in imputed geotags when using region specific Twitter word counts as model features. This is the first work to use tweets with imputed geotags in subsequent forecasting analyses and the first attempt to understand the impact on model performance in doing so.

Transitioning the dynamic logistic regression model to one with Berkson measurement error allows the uncertainty in the imputed geotag to be used as model information, effectively including all sampled tweets in further analyses. The Poisson-Binomial error structure uses the uncertainty as a measure of how likely the tweet originated from the region of interest. The more density placed in the region by the hybrid geotag imputation model, the more probable it is the origin. Therefore, even data sets where few imputed geotags fall within the region of interest can provide forecasting information to the model.

The simulation study showed comparable results between the fixed covariate model and the measurement error model. In the case where each model has an adequate amount of data, the measurement error model shows equivalent forecast performance. In the application of forecasting influenza, the forecast results were similar for the largest region sizes. For example, at the national level, results were identical, and in the South Atlantic region, forecast performance varied slightly by the forecast period.

As the region size for the geographic model reduced to a single state, we showed the reduction in exploitable data renders the fixed covariate model much less effective, while the measurement error model maintains similar performance across regions. Out of the 100,959 tweets collected not containing an origin, only 44 imputed geotags were in North Carolina. However, the hybrid geotag imputation model calculated approximately 1,322 tweets that might have originated in the state. The measurement error model is able to use this uncertainty information to maintain accurate forecast estimates.

Chapter 5

Concluding Remarks and Future Work

The goal of this research is to develop novel statistical methodology enabling the use of social media as a tool for event detection and enrichment. In this dissertation we focused exclusively on enhancing the value of Twitter and its predictive capability. Motivated by applications of influenza and civil unrest, we explored the areas of geotag imputation, short-term Bernoulli response forecasting using dynamic logistic regression, scalable dynamic variable selection, and measurement error of geographic specific models caused by location uncertainty in tweets. Below we summarize the contributions presented in this dissertation and future work directions.

In Chapter 2, we improved geotgag imputation according to prediction accuracy, precision, and coverage. This is the second work to propose a hybrid geotag imputation algorithm and the first to consider using model features simultaneously. Unlike Rahimi et al. [2015], we use the tweet text and user network information jointly to predict the origin of tweets. To do so, we weight the estimate towards the geographically narrow features which allows the model to determine the most predictive components. We showed according to the popular Eisentein data set [Eisenstein et al., 2010], the proposed hybrid model imputes geotags to within 19 km at least 50% of the time, nearly an 8-fold increase in accuracy compared to Rahimi et al. [2015].

Furthermore, the hybrid model of Chapter 2 is an extension of Priedhorsky et al. [2014]. We also use Gaussian mixture models to quantify the uncertainty of the location estimate. This approach provides a visual interpretation of model confidence as the imputed geotag is a probability distribution across the earth's surface estimating each set of coordinates (latitude × longitude) being the origin of the tweet. Using both text and network information as model features, we show improved precision and more accurate coverage compared to Priedhorsky et al. [2014], the only other researcher to quantify uncertainty.

A natural extension of geotag imputation is to consider the problem of spatial event enrichment. That is, do tweets of similar content that cluster within narrow geographic areas reveal unknown information? Tweet clustering has been examined within the context of the London riots [Thom et al., 2012], Hurricane Irene [Thom et al., 2012], and the Japan Tsunami [Sakaki et al., 2010]. In each case, Twitter was used as an event enrichment tool to determine locations most in need of aid. Researchers have yet to consider using tweets with imputed geotags for event enrichment. The challenge becomes clustering tweets with and without location uncertainty. One could extend the work clustering bivariate Gaussian observations [Kumar and Patel, 2007] to Gaussian mixture model distributed observations in order to utilize the methodology of geotag imputation in Chapter 2 for event enrichment.

Another future work area, with particular interest to the nonproliferation community, is the investigation of spatial event detection. That is, do tweets of similar content that cluster within narrow geographic areas reveal an unknown event? For example, is it possible to uncover a local event oblivious to the media and the masses in general, such as a nuclear power plant acting irregularly? Detecting such an event, even days or hours prior to a national news story, may stop acts of nuclear material diversion. Given that the event is unknown, statistical methods would likely include confidence estimates in the likelihood of an event having occurred.

In Chapter 3 we improved the forecasting and variable selection accuracy of civil unrest in Latin America. We showed the dynamic logistic regression model is a better forecast tool than the static baseline. Unlike static logistic regression, the dynamic model is able to account for dependencies in successive observations. The dynamic model is able to evolve in time as new information becomes available and therefore, forecasts according to the most recent data. Although a powerful forecasting tool for Bernoulli response time series, dynamic logistic regression literature includes only a single variable selection technique. Extending the work of Bondell and Reich [2012], we used penalized credible regions to conduct variable selection dynamically, or for each state vector $\boldsymbol{\beta}_t$ in time. This approach is scalable compared to Dynamic Model Averaging [McCormick et al., 2012] and outperformed static logistic regression with LASSO regularization according to precision, recall, and F1 scores in simulation and application. The variable selection approach allowed us to contextualize the reasons for protest in each of the six Latin American countries considered.

Given the success of forecasting civil unrest with social media data, an obvious future direction is to implement the methodology in different applications. Similar to Zhang et al. [2011], one could use Twitter to aggregate the emotional state of the economy in order to forecast stock market indicators such as the DASDAQ. Instead, we could use dynamic logistic regression to forecast the probability of the stock market indicator closing with a gain or loss for the day. Other possible applications using dynamic logistic regression and Twitter data as model features include forecasting crime [Gerber, 2014], dengue fever [Gomide et al., 2011], and election polls [Tumasjan et al., 2010], to name a few.

Although the penalized credible region approach performed well for variable selection, additional techniques should be explored. A natural next step is to consider shrinkage priors of the Bayesian static and dynamic linear model literature. Given the pseudo-Gaussian data provided by the inclusion of the Pólya-Gamma latent variable, shrinkage priors such as the Bayesian LASSO Laplace prior [Park and Casella, 2008] or Normal-Gamma prior [Griffin et al., 2010] may be applied to the dynamic logistic regression state vectors. Furthermore, shrinking elements of $\boldsymbol{\beta}_t$ before applying the penalized credible region method of Chapter 3 may lead to better overall variable selection performance as noise variables are shrunk closer to zero.

In Chapter 4, we demonstrated the impact of measurement error in covariates for the dynamic logistic regression model. To our knowledge, this is the first work to use tweets with imputed geotags in a predictive model. We showed that accounting for the location uncertainty in tweets improves model performance when forecasting within small geographic regions. To account for the location uncertainty in the estimate of daily word counts, we assumed a Poisson-Binomial distribution for the measurement error in the covariates, which is a function of the hybrid geotag imputation model of Chapter 2. With the substantial increase in model parameters used to account for the location uncertainty, we opted to apply Approximate Bayesian Computation. This simulation technique allows us to maintain the hybrid model fitting methodology of Chapter 2 and ease the posterior simulation time to estimate the dynamic parameters.

The methodology presented in Chapter 4 is merely a first step in understanding measurement error in social media analytics. Moving forward, there is a great deal of work needed to connect geotag imputation to forecasting methodology. Measurement error in covariates should be considered in more simplistic models shown to be effective in the literature, along with alternative error structures to improve computation time. Alternative simulations should be considered to more fully explore the performance of measurement error models at varying granularities of region size. Regardless of future directions, the methodology of Chapter 2 provides a solid foundation for considering measurement error when forecasting with Twitter, as the hybrid geotag imputation model accurately accounts for uncertainty using a spatial probability distribution.

Connecting geotag imputation to forecasting methodology allows researchers to utilize the majority of tweets which do not contain an origin. By accounting for location uncertainty and effectively increasing the supply of exploitable data, researchers will be able to forecast models at finer geographic region sizes. Instead of modeling disease or civil unrest at the country level, perhaps the granularity could be reduced to the state or city level, providing further insight pertinent to allocating government resources.

BIBLIOGRAPHY

Satyen Abrol and Latifur Khan. Tweethood: Agglomerative Clustering on Fuzzy K-Closest Friends with Variable Depth for Location Mining. In *2010 IEEE Second International Conference on Social Computing (SocialCom)*, pages 153–160. IEEE, 2010.

Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. Predicting Flu Trends Using Twitter Data. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 702–707. IEEE, 2011.

Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. Twitter Improves Seasonal Influenza Prediction. In *Healthinf*, pages 61–70, 2012.

Shruti Aggarwal and Janpreet Singh. Outlier Detection Using K-Mean and Hybrid Distance Technique on Multi-Dimensional Data Set. *International Journal of Advanced Research in Computer Engineering and Technology*, 2(9):2626–31, 2013.

Oluwaseun Ajao, Jun Hong, and Weiru Liu. A Survey of Location Inference Techniques on Twitter. *Journal of Information Science*, 41(6):855–864, 2015.

Hirotugu Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

Daniele Angelosante, Georgios B Giannakis, and Emanuele Grossi. Compressed Sensing of Time-Varying Signals. In *2009 16th International Conference on Digital Signal Processing*, pages 1–8. IEEE, 2009.

Marta Arias, Argimiro Arratia, and Ramon Xuriguera. Forecasting with Twitter Data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):8, 2013.

Sitaram Asur and Bernardo A Huberman. Predicting the Future with Social Media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 492–499. IEEE, 2010.

Lars Backstrom, Eric Sun, and Cameron Marlow. Find Me If You Can: Improving Geographical Prediction With Social and Spatial Proximity. In *Proceedings of the 19th International Conference on World Wide Web*, pages 61–70. ACM, 2010.

Ole E Barndorff-Nielsen and Neil Shephard. Non-Gaussian Ornstein–Uhlenbeck-Based Models and Some of Their Uses in Financial Economics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):167–241, 2001.

BBC. Colombian Farmers Protest Against Government's 'Broken Promises', 2014. URL `http://www.bbc.com/news/world-latin-america-27198890`. [Online; accessed 28-October-2017].

Mark A Beaumont. Approximate Bayesian Computation in Evolution and Ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41:379–406, 2010.

Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate Bayesian Computation in Population Genetics. *Genetics*, 162(4):2025–2035, 2002.

Miguel AG Belmonte, Gary Koop, and Dimitris Korobilis. Hierarchical Shrinkage in Time-Varying Parameter Models. *Journal of Forecasting*, 33(1):80–94, 2014.

Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter Mood Predicts the Stock Market. *Journal of Computational Science*, 2(1):1–8, 2011.

Howard D Bondell and Brian J Reich. Consistent High-Dimensional Bayesian Variable

Selection Via Penalized Credible Regions. *Journal of the American Statistical Association*, 107(500):1610–1624, 2012.

Paola Bortot, Stuart G Coles, and Scott A Sisson. Inference for Stereological Extremes. *Journal of the American Statistical Association*, 102(477):84–92, 2007.

Catherine Calder, Michael Lavine, Peter Müller, and James S Clark. Incorporating Multiple Sources of Stochasticity into Dynamic Population Models. *Ecology*, 84(6):1395–1402, 2003.

Emmanuel Candes and Terence Tao. The Dantzig Selector: Statistical Estimation When p is Much Larger Than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

François Caron, Luke Bornn, and Arnaud Doucet. Sparsity-Promoting Bayesian Dynamic Linear Models. *arXiv preprint arXiv:1203.0106*, 2012.

Raymond J Carroll, David Ruppert, Leonard A Stefanski, and Ciprian M Crainiceanu. *Measurement Error in Nonlinear Models: A Modern Perspective*. CRC Press, 2006.

Chris K Carter and Robert Kohn. On Gibbs Sampling for State Space Models. *Biometrika*, 81(3):541–553, 1994.

CDC. 2009-2010 Influenza Season Week 9 Ending March 6, 2010. `http://www.cdc.gov/flu/weekly/weeklyarchives2009-2010/weekly09.htm`, 2010. [Online; accessed 07-December-2016].

CDC. FluView: National, Regional, and State Level Outpatient Illness and Viral Surveillance, 2017. URL `https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html`. [Online; accessed 09-December-2017].

Swarup Chandra, Latifur Khan, and Fahad Bin Muhaya. Estimating Twitter User Location Using Social Interactions – A Content Based Approach. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom)*, pages 838–843. IEEE, 2011.

Feng Chen and Daniel B Neill. Non-Parametric Scan Statistics for Event Detection and Forecasting in Heterogeneous Social Media Graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1166–1175. ACM, 2014.

Sean X Chen and Jun S Liu. Statistical Applications of the Poisson-Binomial and Conditional Bernoulli Distributions. *Statistica Sinica*, pages 875–892, 1997.

Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You Are Where You Tweet: A Content Based Approach to Geo-Locating Twitter Users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 759–768. ACM, 2010.

Chiogna, Monica Carlo Gaetan, and Carlo Gaetan. Dynamic Generalized Linear Models with Application to Environmental Epidemiology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 51(4):453–468, 2002.

CNN. Colombian President Deploys 50,000 Troops After Violent Protests, 2013. URL `http://www.cnn.com/2013/08/30/world/americas/colombia-protests/index.html`. [Online; accessed 28-October-2017].

Ryan Compton, David Jurgens, and David Allen. Geotagging One Hundred Million

Twitter Accounts with Total Variation Minimization. In *2014 IEEE International Conference on Big Data*, pages 393–401. IEEE, 2014.

Aron Culotta. Towards Detecting Influenza Epidemics By Analyzing Twitter Messages. In *Proceedings of the First Workshop on Social Media Analytics*, pages 115–122. ACM, 2010.

Clodoveu A Davis Jr, Gisele L Pappa, Diogo Rennó Rocha de Oliveira, and Filipe de L Arcanjo. Inferring the Location of Twitter Messages Based on User Relationships. *Transactions in GIS*, 15(6):735–751, 2011.

Petros Dellaportas, Jonathan J Forster, and Ioannis Ntzoufras. On Bayesian Model and Variable Selection Using MCMC. *Statistics and Computing*, 12(1):27–36, 2002.

Darrell Duffie, Leandro Saita, and Ke Wang. Multi-Period Corporate Default Prediction with Stochastic Covariates. *Journal of Financial Economics*, 83(3):635–665, 2007.

Maeve Duggan and Joanna Brenner. *The Demographics of Social Media Users, 2012*, volume 14. Pew Research Center's Internet & American Life Project Washington, DC, 2013.

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least Angle Regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. A Latent Variable Model for Geographic Lexical Variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287. Association for Computational Linguistics, 2010.

Jianqing Fan and Runze Li. Variable Selection Via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, 96(456): 1348–1360, 2001.

Paul Fearnhead and Dennis Prangle. Semi-Automatic Approximate Bayesian Computation. *Arxiv preprint arXiv*, 1004, 2010.

Ingo Feinerer, Kurt Hornik, and David Meyer. Text Mining Infrastructure in R. *Journal of Statistical Software*, 25(5):1–54, March 2008. URL `http://www.jstatsoft.org/v25/i05/`.

Manuel Fernández and Stuart Williams. Closed-Form Expression for the Poisson-Binomial Probability Density Function. *IEEE Transactions on Aerospace and Electronic Systems*, 46(2):803–817, 2010.

Marco Ferreira and Dani Gamerman. Dynamic generalized linear models. In *Generalized Linear Models: A Bayesian Perspective*, pages 57–72, New York, Marcel Dekker, 2000.

Chris Fraley, Adrian E. Raftery, Thomas Brendan Murphy, and Luca Scrucca. Mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation. Technical Report 597, Department of Statistics, University of Washington, 2012.

David T Frazier, Christian P Robert, and Judith Rousseau. Model Misspecification in ABC: Consequences and Diagnostics. *arXiv preprint arXiv:1708.01974*, 2017.

Dani Gamerman. Markov Chain Monte Carlo for Dynamic Generalised Linear Models. *Biometrika*, 85(1):215–227, 1998.

Judith Gelernter and Nikolai Mushegian. Geo-parsing Messages from Microtext. *Transactions in GIS*, 15(6):753–773, 2011.

Nicholas Generous, Geoffrey Fairchild, Alina Deshpande, Sara Y Del Valle, and Reid Priedhorsky. Global Disease Monitoring and Forecasting with Wikipedia. *PLoS Computational Biology*, 10(11):e1003892, 2014.

Edward I George and Robert E McCulloch. Variable Selection Via Gibbs Sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

Matthew S Gerber. Predicting Crime Using Twitter and Kernel Density Estimation. *Decision Support Systems*, 61:115–125, 2014.

Janaína Gomide, Adriano Veloso, Wagner Meira Jr, Virgílio Almeida, Fabrício Benevenuto, Fernanda Ferraz, and Mauro Teixeira. Dengue Surveillance Based on a Computational Model of Spatio-Temporal Locality of Twitter. In *Proceedings of the 3rd International Web Science Conference*, page 3. ACM, 2011.

Aude Grelaud, Christian P Robert, Jean-Michel Marin, François Rodolphe, Jean-François Taly, et al. ABC Likelihood-Free Methods for Model Choice in Gibbs Random Fields. *Bayesian Analysis*, 4(2):317–335, 2009.

Jim E Griffin, Philip J Brown, et al. Inference With Normal-Gamma Prior Distributions in Regression Problems. *Bayesian Analysis*, 5(1):171–188, 2010.

Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H Chi. Tweets From Justin Bieber's Heart: The Dynamics of the Location Field in User Profiles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 237–246. ACM, 2011.

Kyle S Hickmann, Geoffrey Fairchild, Reid Priedhorsky, Nicholas Generous, James M Hyman, Alina Deshpande, and Sara Y Del Valle. Forecasting the 2013–2014 Influenza Season Using Wikipedia. *PLoS Computational Biology*, 11(5):e1004239, 2015.

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian Model Averaging: A Tutorial. *Statistical Science*, 14(4):382–401, 1999.

Yili Hong. On Computing the Distribution Function for the Poisson Binomial Distribution. *Computational Statistics & Data Analysis*, 59:41–51, 2013.

Yili Hong, William Q Meeker, and James D McCalley. Prediction of Remaining Life of Power Transformers Based on Left Truncated and Right Censored Lifetime Data. *The Annals of Applied Statistics*, 3(2):857–879, 2009.

Bernardo A Huberman, Daniel M Romero, and Fang Wu. Social Networks That Matter: Twitter Under the Microscope. *arXiv preprint arXiv:0812.1045*, 2008.

Mans Hulden, Miikka Silfverberg, and Jerid Francom. Kernel Density Estimation for Text-Based Geolocation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 145–150, 2015.

David Jurgens. That's What Friends Are For: Inferring Location in Online Social Media Platforms Based on Social Relationships. *ICWSM*, 13:273–282, 2013.

Maria Kalli and Jim E Griffin. Time-Varying Sparsity in Dynamic Regression Models. *Journal of Econometrics*, 178(2):779–793, 2014.

Robert E Kass and Adrian E Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

Sheila Kinsella, Vanessa Murdock, and Neil O'Hare. I'm Eating a Sandwich in Glasgow: Modeling Locations With Tweets. In *Proceedings of the 3rd International Workshop on Search and Mining User-Generated Contents*, pages 61–68. ACM, 2011.

Gary Koop and Dimitris Korobilis. Forecasting Inflation Using Dynamic Model Averaging. *International Economic Review*, 53(3):867–886, 2012.

Gizem Korkmaz, Jose Cadena, Chris J Kuhlman, Achla Marathe, Anil Vullikanti, and Naren Ramakrishnan. Combining Heterogeneous Data Sources for Civil Unrest Forecasting. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 258–265. ACM, 2015.

Mahesh Kumar and Nitin R Patel. Clustering Data with Measurement Errors. *Computational Statistics & Data Analysis*, 51(12):6084–6101, 2007.

Kathy Lee, Ankit Agrawal, and Alok Choudhary. Real-Time Disease Surveillance Using Twitter Data: Demonstration on Flu and Cancer. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1474–1477. ACM, 2013.

Jiwei Li and Claire Cardie. Early Stage Influenza Detection From Twitter. *arXiv preprint arXiv:1309.7340*, 2013.

Jinchi Lv and Yingying Fan. A Unified Approach to Model Selection and Sparse Recovery Using Regularized Least Squares. *The Annals of Statistics*, 37(6a):3498–3528, 2009.

Richard Mallinson. *The Arab Uprisings and MENA Political Instability–Implications for Oil & Gas Markets.* Oxford Institute for Energy Studies, 2014.

Tyler H McCormick, Adrian E Raftery, David Madigan, and Randall S Burd. Dynamic Logistic Regression and Dynamic Model Averaging for Binary Classification. *Biometrics*, 68(1):23–30, 2012.

Jeffrey McGee, James Caverlee, and Zhiyuan Cheng. Location Prediction in Social Media Based on Tie Strength. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 459–468. ACM, 2013.

Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M Carley. Is the Sample Good Enough? Comparing Data From Twitter's Streaming API with Twitter's Firehose. *arXiv preprint arXiv:1306.5204*, 2013.

Jouchi Nakajima and Mike West. Bayesian Analysis of Latent Threshold Dynamic Models. *Journal of Business & Economic Statistics*, 31(2):151–164, 2013.

John A Nelder and R Jacob Baker. Generalized Linear Models. *Encyclopedia of Statistical Sciences*, 1972.

Atushi Niida, Seiya Imoto, Teppei Shimamura, and Satoru Miyano. Statistical Model-Based Testing to Evaluate the Recurrence of Genomic Aberrations. *Bioinformatics*, 28(12):i115–i120, 2012.

Trevor Park and George Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

Giovanni Petris, Sonia Petrone, and Patrizia Campagnoli. Dynamic Linear Models. In *Dynamic Linear Models with R*. Springer-Verlag New York, 2009.

Ermanno Pitacco. Mortality and Longevity: A Risk Management Perspective. In *Invited lecture at the 1st IAA Life Colloquium, Stockholm*, 2007.

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

Martin F Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.

Reid Priedhorsky, Aron Culotta, and Sara Y Del Valle. Inferring the Origin Locations of Tweets with Quantitative Confidence. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1523–1536. ACM, 2014.

Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. Population Growth of Human Y Chromosomes: A Study of Y Chromosome Microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 1999.

Adrian E Raftery, Miroslav Kárnỳ, and Pavel Ettler. Online Prediction Under Model Uncertainty Via Dynamic Model Averaging: Application to a Cold Rolling Mill. *Technometrics*, 52(1):52–66, 2010.

Afshin Rahimi, Duy Vu, Trevor Cohn, and Timothy Baldwin. Exploiting Text and Network Context for Geolocation of Social Media Users. *arXiv preprint arXiv:1506.04803*, 2015.

Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, et al. 'Beating the News' with EMBERS: Forecasting Civil Unrest Using Open Source Indicators. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1799–1808. ACM, 2014.

Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldridge. Supervised Text-Based Geolocation Using Language Models on an Adaptive Grid. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1500–1510. Association for Computational Linguistics, 2012.

Dominic Rout, Kalina Bontcheva, Daniel Preoţiuc-Pietro, and Trevor Cohn. Where's@ Wally?: A Classification Approach to Geolocating Users Based on Their Social Ties. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 11–20. ACM, 2013.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake Shakes Twitter Users: Real-Time Event Detection By Social Sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860. ACM, 2010.

Axel Schulz, Aristotelis Hadjakos, Heiko Paulheim, Johannes Nachtwey, and Max Mühlhäuser. A Multi-Indicator Approach for Geolocalization of Tweets. In *ICWSM*, 2013.

Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2): 461–464, 1978.

Dino Sejdinovic, Christophe Andrieu, and Robert Piechocki. Bayesian Sequential Compressed Sensing in Sparse Dynamical Systems. In *48th Annual Allerton Conference on Communication, Control and Computing*. IEEE, 2010.

Scott A Sisson, Yanan Fan, and Mark M Tanaka. Sequential Monte Carlo Without Likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.

Mikael Sunnåker, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, and Christophe Dessimoz. Approximate Bayesian Computation. *PLoS Computational Biology*, 9(1):e1002803, 2013.

Mark M Tanaka, Andrew R Francis, Fabio Luciani, and SA Sisson. Using Approximate Bayesian Computation to Estimate Tuberculosis Transmission Parameters from Genotype Data. *Genetics*, 173(3):1511–1520, 2006.

Dennis Thom, Harald Bosch, Steffen Koch, Michael Wörner, and Thomas Ertl. Spatiotemporal Anomaly Detection Through Visual Analysis of Geolocated Twitter Messages. In *2012 IEEE Pacific Visualization Symposium (PacificVis)*, pages 41–48. IEEE, 2012.

Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the Number of Clusters in a Data Set Via the Gap Statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and Smoothness Via the Fused Lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

TIME. The 500 Most Frequently Used Words on Twitter. `http://techland.time.com/2009/06/08/the-500-most-frequently-used-words-on-twitter/`, 2009. [Online; accessed 07-December-2016].

Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael PH Stumpf.

Approximate Bayesian Computation Scheme for Parameter Inference and Model Selection in Dynamical Systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.

Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting Elections With Twitter: What 140 Characters Reveal About Political Sentiment. *ICWSM*, 10(1):178–185, 2010.

Brandon M Turner and Trisha Van Zandt. Hierarchical Approximate Bayesian Computation. *Psychometrika*, 79(2):185–209, 2014.

Mike West, P Jeff Harrison, and Helio S Migon. Dynamic Generalized Linear Models and Bayesian Forecasting. *Journal of the American Statistical Association*, 80(389): 73–83, 1985.

Mike West, Raquel Prado, and Andrew D Krystal. Evaluation and Comparison of EEG Traces: Latent Structure in Nonstationary Time Series. *Journal of the American Statistical Association*, 94(446):375–387, 1999.

Jesse Windle, Carlos M Carvalho, James G Scott, and Liang Sun. Efficient Data Augmentation in Dynamic Models for Binary and Count Data. *arXiv preprint arXiv:1308.0774*, 2013.

Benjamin Wing and Jason Baldridge. Hierarchical Discriminative Classification for Text-Based Geolocation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 336–348. Association for Computational Linguistics, 2014.

Benjamin P Wing and Jason Baldridge. Simple Supervised Document Geolocation with

Geodesic Grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 955–964. Association for Computational Linguistics, 2011.

Ming Yuan and Yi Lin. Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Xue Zhang, Hauke Fuehres, and Peter A Gloor. Predicting Stock Market Indicators Through Twitter "I Hope It Is Not As Bad As I Fear". *Procedia-Social and Behavioral Sciences*, 26:55–62, 2011.

Hui Zou. The Adaptive Lasso and its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

Hui Zou and Trevor Hastie. Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.