Predictive Model Fusion:
A Modular Approach to Big, Unstructured Data

Andrew Blake Hoegh

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Statistics

Scotland Leman, Chair
Marco Ferreira
Dave Higdon
Naren Ramakrishnan

March 22, 2016
Blacksburg, Virginia

Keywords: Model Fusion, Spatiotemporal Modeling, Areal Data, Sequential Monte Carlo

Predictive Model Fusion:
A Modular Approach to Big, Unstructured Data Problems

Andrew Blake Hoegh

Data sets of increasing size and complexity require new approaches for prediction as the sheer volume of data from disparate sources inhibits joint processing and modeling. Rather modular segmentation is required, in which a set of models process (potentially overlapping) partitions of the data to independently construct predictions. This framework enables individuals models to be tailored for specific selective superiorities without concern for existing models, which provides utility in cases of segmented expertise. However, a method for fusing predictions from the collection of models is required as models may be correlated. This work details optimal principles for fusing binary predictions from a collection of models to issue a joint prediction. An efficient algorithm is introduced and compared with off the shelf methods for binary prediction. This framework is then implemented in an applied setting to predict instances of civil unrest in Central and South America. Finally, model fusion principles of a spatiotemporal nature are developed to predict civil unrest. A novel multiscale modeling is used for efficient, scalable computation for combining a set of spatiotemporal predictions.

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

The authors acknowledge Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this dissertation.

Dedication

To Eleanor and Georgiana.

Chase your dreams.

Acknowledgments

Graduate school and the dissertation process is daunting. I'd like to thank my advisor Scotland Leman for his support and guidance these last five years. I now have a high standard to live up to as an advisor. I've also been lucky to have Marco Ferreira as an unofficial second advisor who has played a large part in my success. I'd also like to thank all of my fellow students that helped me on this journey, including many who spent every Friday afternoon together.

Countless other people across Virginia Tech community have been instrumental in my success. Naren Ramakrishnan and Dave Higdon have been heavily involved in this dissertation and in my development as a scholar. Eric Vance and Chris Franck's guidance in LISA gave me an introduction to research and instilled valuable lessons for my professional growth. Leanna House for encouragement and advice. Eric Smith for my development as a teacher. Marcos Carzolio has been helpful in developing ideas and providing enough humor to keep this journey interesting.

I want to thank my parents for unending encouragement. Emma thank you for allowing me to chase my dream. You have been my support system and this does not happen without you. I cannot wait for the next chapter of our life in the Big Sky Country of Bozeman, Montana. It has truly been a pleasure to be part of the Blacksburg and Virginia Tech community. LETS GO... HOKIES!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This research is motivated by predicting civil unrest as part of a large interdisciplinary team which includes political scientists, computer scientists, mathematicians, and statisticians. Civil unrest is a sufficiently complicated phenemenon with a vast set of heterogenous data that a comprehensive model jointly considering all relevant data sources is infeasible. Rather a modular approach is implemented using a variety of algorithms developed independently to extract information from heterogenous datasets. This approach allows distinct groups (and expertise) to develop predictive algorithms separately, which can result in selective superiorities where each model will have distinct strengths and weaknesses. While the algorithms are developed independently, they are not guaranteed to be statistically independent. Algorithms may use overlapping data sources or capture similar signals inducing correlation in the algorithm predictions. The statistical research question and the focus of this dissertation is on integrating, or fusing, these predictions in a principled manner. Though our applications will focus on modeling civil unrest, the theory and methods presented here are completely general and will apply of modular prediction to data problems: both big and small.

Given the complexity of model fusion and the intricacies of our specific application, there are numerous theoretical and applied challenges. We focus on principles for model fusion, an application of optimal fusion techniques for binary data, and efficient, scalable spatiotemporal modeling

framework implementing model fusion using multiscale modeling. The remainder of this chapter reviews existing methodology that provides the foundation for the contributions of this research.

We initially review various concepts related to model fusion including ensemble methods and model averaging. Then given that civil unrest is inherently a spatiotemporal phenomenon, we consider methods that account for the spatial and temporal effects in the data. In particular, we provide an overview of basic spatiotemporal methods with an emphasis on areal data. As we implement our modeling from a Bayesian viewpoint throughout, we also detail method for Bayesian computation with a focus on spatiotemporal modeling. In addition to the literature review in the introduction section, each of the following chapters 2, 3, and 4 are self contained manuscripts that contain an overview of related methods.

1.1 Fusion Concepts

Broadly speaking, model fusion is a process that combines output from multiple models. In this work we focus on predictive model fusion where the goal is prediction; however, inferential model fusion is also a developing research area, particularly with the push to extract meaningful information from *big data*. While general model fusion principles are applicable to any type of data, the data for predicting civil unrest are either binary or count data, so we restrict our focus to methods for these data types.

1.1.1 Ensemble Methods

In spirit, model fusion is similar to ensemble methods although there is a clear difference. With model fusion we condition on a fixed set of models (or algorithms); whereas, many ensemble methods are designed to construct a series of models. Nevertheless, we discuss ensemble methods and the principles contained therein. One popular method for constructing and combining a series of classifiers is bootstrap aggregating, or bagging, (Breiman, 1996). The bagging procedure

samples the dataset (with replacement) many times to create a set of datasets. A classifier, typically a decision tree, is then applied to each of the created subsets and predictions are uniformly averaged across the sets. A very similar method known as random forests was also developed by Breiman (2001). In a random forest algorithm a set of decision trees are applied to randomly sampled datasets, as in boosting; however, the random forest also uses a random sample of covariates for constructing each split in a decision tree. In a slightly different vein, an ensemble procedure known as boosting (Schapire, 1990; Freund and Schapire, 1997) is also effective for constructing and combining a series of classifiers. The most common implementation of boosting is the AdaBoost algorithm, which has gained the moniker as the *best out-of-the-box classifier*. Essentially the AdaBoost algorithm learns weights for a set of classifiers in an iterative fashion. A nice elementary explanation of AdaBoost is given by Rojas (2009).

A major challenge is that while the models are independently constructed, overlap in datasets exists which can lead to correlated model output. Hence, we are interested in principles for combining models when correlation is present, which none of these ensemble methods incorporate.

1.1.2 Predictive Model Fusion

Using different terminology, Wolpert (1992) provides an overview of the model fusion problem, focusing on ways to combine generalizers (models) for prediction. Given a set of three continuous model predictions Venkataramani and Kumar (2006) details the optimal fusion for binary output. From a different perspective by conditioning on the the models, then model fusion is similar to traditional regression or classification problems with a specific set of covariates. In this framework, we consider Linear Discriminate Analysis (LDA) and Quadratic Discriminate Analysis (QDA), (see (Hastie et al., 2009) for example). LDA and QDA are methods to learn a discriminate function, or decision rule. Both LDA and QDA learn $P[Y = i|\boldsymbol{X}]$, that is the probability of class i given data $\boldsymbol{X}$. The difference is that QDA permits different covariance structures for classes i and j, which enables non-linear boundaries in the discriminate function. LDA and QDA are both designed for continuous (Gaussian) data in order to issue binary probabilities. Our fusion method, presented in

chapter 2 extends this to the binary case.

1.1.3 Bayesian Model Averaging

Another method often used to combine multiple models is Bayesian Model Averaging (Hoeting et al., 1999). Predictions from Bayesian model averaging are weighted by posterior model probabilities as

$$\sum_k \int P(Y_*|\theta_k, M_k, \mathcal{D})p(\theta_k|M_k, \mathcal{D})p(M_k|\mathcal{D})d\theta_k.$$

MCMC stochastic search algorithms are typically employed so explore the model space. Popular methods include George and McCulloch (1993)'s stochastic search and variable selection, Madigan et al. (1995) Markov chain Monte Carlo model composition, and more recently Bottolo and Richardson (2010)'s evolutionary stochastic search. These algorithms all use different techniques to explore the model space with the intent of learning $P(M_k|\mathcal{D})$; however, they do not explicitly capture correlation between models.

1.2 Spatiotemporal Models

Next we discuss spatiotemporal methods, which are necessary to model the process by which instances of civil unrest propagate and cascade through space and time. This section reviews spatial methods with a focus on areal data and time series methods. We then discuss relevant spatiotemporal methods.

1.2.1 Spatial Methods

Spatial relationships typically follow the so called first law of geography: "Everything is related to everything else, but near things are more related than distant things" (Tobler, 1970). The challenge in fitting spatial models is in defining 'near.' For point referenced data, this concept is fairly clear.

Nearness is defined in terms of Euclidean distance and covariance functions, such as the Matérn (1960) class, are defined by the change in correlation as the Euclidean distance between points changes.

For areal data, responses are aggregated on some predefined region such as a city or a ZIP code. In this case, the concept of nearness isn't inherently clear. Euclidean distance can still be used, the points to use for the distance calculation are not uniquely defined. Centroids could be used, or nearest points, or farthest points. Typically a different approach is taken with areal data in which nearness is defined through the concept of neighboring regions. Given a set of neighbors, a typical assumption is that of a Markov random field. In a Markov random field, given the values of the neighboring regions the observation for a specified locale is conditionally independent from the remaining regions. Formally, $P(y_i|y_j, j \neq i) = P(y_i|y_j, j \in \mathcal{N}(i))$, where $\mathcal{N}(i)$ denotes neighbors of region i.

The conditional dependence structure is exploited by Besag (1974) in what were called 'auto' models. These models are equivalent to the popular Conditional AutoRegressive (CAR) models (Besag and Kooperberg, 1995). A benefit of this conditional independence structure is easier MCMC algorithms (Besag and Green, 1993; Besag et al., 1995). CAR models have become increasingly popular and used in many applications, see (Waller et al., 1997; Carlin and Banerjee, 2003) for example. Despite the conditional independence structure, computation for CAR models and the resulting MCMC methods becomes increasingly difficult as the size of the spatial data set increases. This is magnified when spatial effects are present across several hierarchical layers of the data, such as city/state/country. This framework would require computing CAR models across three levels.

A growing research area, particularly for spatial data with a hierarchical structure are multiscale methods (Ferreira and Lee, 2007). Up to this point most of the research has been focused on point referenced data of Gaussian areal data (Ferreira et al., 2010, 2011). Fonseca and Ferreira (2016) recently developed a method for handling spatiotemporal Poisson data. In this method, covariates vary temporally, but only vary spatially at the coarsest level. As a result, covariates at

lower levels of the hierarchy cannot be handled in this model. Chapter 4 outlines a novel multiscale spatiotemporal that allows covariates to vary spatially across levels of the hierarchy. Additionally, this novel implementation has easy extensions to modeling multivariate counts, which is outlined in the discussion.

1.2.2 Time Series Methods

State space models are a class of models that have been proven to be useful for modeling time series data as well as spatiotemporal data. They typically include a latent term that evolves temporally. A popular class of state space models are Dynamic Linear Models (DLMs) as detailed in West and Harrison (1997). Formally, DLMs are implemented with the following setup:

$$\begin{aligned} \text{Observation Equation:} \quad & y_t = F'x_t + \nu_t, \qquad \nu_t \sim N(0, V), \\ \text{State Equation:} \quad & x_t = Gx_{t-1} + \omega_t, \quad \omega_t \sim N(0, W_t). \end{aligned}$$

Although, the error structure need not be normally distributed. DLMs incorporate classical time series methods from the ARIMA paradigm (Box and Pierce, 1970; Box, 1994) as well as filtering methods such as the Kalman filter (Kalman, 1960). Furthermore, the error terms need not be independent, but rather, can be used to induce spatial correlation (see for example Hoegh and Leman (2014).) Futhermore, DLMs can be used to model the temporal dynamics within a spatiotemporal multiscale framework.

In most cases, spatial and temporal effects are assumed to separable in which the covariance can be factored as the product of the spatial and temporal components. In other words, the spatial correlation does not change with respect to time. For this work, we restrict our focus to separable spatiotemporal covariance structures; although, future work will address dynamic spatial structures within a multiscale framework.

Spatiotemporal Fusion

Spatiotemporal model fusion is a relatively new phenomenon without a substantial literature. One similar framework to our problem of predicting civil unrest is the disease surveillance field. Rolka et al. (2007) detail many of the challenges inherent in fusing multiple data steams together for public health surveillance, which include misaligned timing and potentially correlated data streams. One such work is Banks et al. (2012) in which a spatiotemporal surveillance model is constructed for syndromic surveillance. While these works are informative, our setting is different and requires new models to capture the spatiotemporal dynamics of civil unrest.

1.3 Bayesian Computation for Spatiotemporal Data

Two common approaches for computation with Bayesian spatiotemporal models are MCMC and Sequential Monte Carlo (SMC). In the case where both the observation equation and evolution equation have known normal errors, the Kalman Filter can be used to compute and exact analytical solution of $P(X_{1:T}|Y_{1:T})$. With unknown normal errors, the Kalman Filter can be embedded in an MCMC procedure for which the state variables are sampled using a procedure known as Forward-Filtering Backward-Sampling (FFBS) (Frühwirth-Schnatter, 1994; Carter and Kohn, 1994). However, we wish to model counts at the observation level and $P(X_{1:T}|Y_{1:T})$ does not have a closed form. When T is large, a joint update of $P(X_{1:T}|Y_{1:T})$ is difficult due to the correlation in $X_{1:T}$. Hence, we use Andrieu et al. (2010)'s Particle-MCMC (PMCMC) for estimation. PMCMC provides a way to update the state-vector in a state space via particle methods embedded in an MCMC framework. One drawback of MCMC methods is that online estimation is not feasible. That is when constructing 1-step ahead predictions the entire sampler needs to be re-run.

In contrast, SMC, permits online computations. Mathematically, this is expressed such that the information in $P(x_t|y_{1:t})$ is used as a prior for learning $P(x_{t+1}|y_{1:t+1})$. With the Kalman equations, the integrals for this transition have analytical solutions. However, in cases where the integrals cannot be analytically computed as procedure known as a particle filter is often implemented using

the following equations:

$$\begin{aligned} p(x_{t+1}|y_{1:t+1}) &\propto p(y_{t+1}|x_{t+1}) \int p(x_{t+1}|x_t)p(x_t|y_{1:t})dx_t \\ &\propto p(y_{t+1}|x_{t+1})p(x_{t+1}|y_{1:t}). \end{aligned}$$

The algorithm for a particle filter is shown in Algorithm 1, where the propagate step computes the

Algorithm 1 Algorithm for Particle Filter (Bootstrap Filter)

1. (Propogate): $\{X_t\}$ to $\{\tilde{X}\}$ via $p(X_{t+1}|X_t)$
2. (Resample): $\{X_{t+1}\}$ with $w_{t+1} \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|x)$

integral in $p(x_{t+1}|y_t)$ and the resample step updates the particles. The issue is this assumes evolution errors (and observation errors if applicable) are known, which is not realistic in most scenarios. Instead we need to use an algorithm capable of learning the unknown parameters, such as particle learning (Carvalho et al. (2010).) Particle Learning, shown in Algorithm 2, makes use of Pitt and Shephard (1999)'s Auxiliary Particle filter, in which the resampling step precedes the propagate step. This has been shown to reduce particle degeneracy by only retaining the most promising particles. The particle learning algorithm updates the parameters jointly with the particles in the sequential Monte Carlo steps. The drawback to the particle learning approach is that sufficient

Algorithm 2 Algorithm for Particle Learning

1. (Resample): $\{\tilde{Z}_t\}$ from $Z_t = (X_t, S_t, \theta)$ with weights $w_t \propto p(y_{t+1}|z_t)$
2. (Propogate): $\tilde{X}$ to X_{t+1} via $p(X_{t+1}|\tilde{Z}_t, y_{t+1})$
3. (Propogate): Sufficient statistics $S_{t+1} = \mathcal{S}(\tilde{S}, X_{t+1}, y_{t+1})$
4. (Resample): θ from $p(\theta|S_{t+1})$.

statistics need to be identified for the parameters. In some sense, parallels can be drawn between particle learning and Gibbs sampling leaving the case where the full conditionals are not a known distribution. This requires another approach, namely the particle filtering procedure dubbed "Liu and West" after Liu and West (2001). Rather than tracking sufficient statistics for fixed parameters, this methodology perturbs the particles for fixed components by drawing them from a mixture of normals. The algorithm is described in Algorithm 3.

Algorithm 3 Algorithm for Liu & West

1. (Identify Prior Point Estimates): $\mu_{t+1}^{(j)} = E(x_{t+1}|x_t^{(j)}, \theta_t^{(j)})$ and $m_t^{(j)} = a\theta_t^{(j)} + (1-a)\bar{\theta}_t$
2. (Sample Indices): k with $w \propto p(y_{t+1}|\mu_{t+1}^{(j)}, m_t^{(j)})$
3. (Sample θ): $\theta_{t+1}^{(k)} \sim N(.|\mu_t^{(k)}, h^2 V_t)$
4. (Propogate x_{t+1}) : $x_{t+1}^{(k)} = p(x_{t+1}^{(k)}|x_t^{(k)}, \theta_{t+1}^{(k)})$
5. (Resample): $\{x_{t+1}, \theta_{t+1}\}^{(k)}$ with $w \propto \frac{p(y_{t+1}|x_{t+1}^{(k)}, \theta_t^{(k)})}{p(y_{t+1}|\mu_{t+1}^{(k)}, m_t^{(k)})}$

Both of these algorithms give satisfactory results when a small number of fixed, unknown parameters are included in the model. In certain cases all of the unknown static parameters will not have sufficient statistics; hence, Niemi (2009) proposes a mix of Particle Learning and the Liu and West filter.

1.4 Dissertation Outline

The remainder of this dissertation contains three manuscripts and outlines extensions and additional work to be pursued in a closing discussion. Chapter 2 presents the theoretical underpinnings of model fusion including optimal strategies. This chapter includes a fusion algorithm for binary data that sample Gaussian Graphical Models for latent probit variables. Chapter 3 highlights an applied solution for model fusion to predict civil unrest. This chapter has been accepted for publication in Technometrics. Chapter 4 introduces a novel spatiotemporal multiscale framework for fusing alerts from the predictive algorithms. This chapter has been accepted for publication in the Journal of the Royal Statistical Society: Series C (Applied Statistics). Chapter 5 summarizes the contributions detailed in Chapter 2 - Chapter 4 and highlights ongoing and future work.

Bibliography

Andrieu, C., Doucet, A., and Holenstein, R. (2010), "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72, 269–342.

Banks, D., Datta, G., Karr, A., Lynch, J., Niemi, J., and Vera, F. (2012), "Bayesian CAR models for syndromic surveillance on multiple data streams: Theory and practice," *Information Fusion*, 13, 105 – 116, special Issue on Information Fusion Applications to Human Health and Safety.

Besag, J. (1974), "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 36, pp. 192–236.

Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995), "Bayesian computation and stochastic systems," *Statistical Science*, 10, pp. 3–41.

Besag, J. and Green, P. J. (1993), "Spatial statistics and Bayesian computation," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 55, pp. 25–37.

Besag, J. and Kooperberg, C. (1995), "On conditional and intrinsic autoregressions," *Biometrika*, 82, 733–746.

Bottolo, L. and Richardson, S. (2010), "Evolutionary stochastic search for Bayesian model exploration," *Bayesian Analysis*, 5, 583–618.

Box, G. (1994), *Time Series Analysis: Forecasting & Control, 3/e*, Pearson Education India.

Box, G. E. and Pierce, D. A. (1970), "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American Statistical Association*, 65, 1509–1526.

Breiman, L. (1996), "Bagging predictors," *Machine Learning*, 24, 123–140.

— (2001), "Random forests," *Machine Learning*, 45, 5–32.

Carlin, B. P. and Banerjee, S. (2003), "Hierarchical multivariate CAR models for spatio-temporally correlated survival data," *Bayesian Statistics*, 7, 45–63.

Carter, C. K. and Kohn, R. (1994), "On Gibbs sampling for state space models," *Biometrika*, 81, 541–553.

Carvalho, C. M., Johannes, M. S., Lopes, H. F., and Polson, N. G. (2010), "Particle learning and smoothing," *Statistical Science*, 25, 88–106.

Ferreira, M. A. R., Bertolde, A. I., and Holan, S. H. (2010), "Analysis of economic data with multiscale spatio-temporal models." *In Handbook of Applied Bayesian Analysis, eds. A. O'Hagan and M. West*.

Ferreira, M. A. R., Holan, S. H., and Bertolde, A. I. (2011), "Dynamic multiscale spatiotemporal models for Gaussian areal data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73, 663–688.

Ferreira, M. A. R. and Lee, H. K. H. (2007), *Multiscale modeling: a Bayesian perspective*, Springer Science & Business Media.

Fonseca, T. C. O. and Ferreira, M. A. R. (2016), "Dynamic multiscale spatiotemporal models for Poisson data," *Journal of the American Statistical Association*, to appear.

Freund, Y. and Schapire, R. E. (1997), "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, 55, 119–139.

Frühwirth-Schnatter, S. (1994), "Data augmentation and dynamic linear models," *Journal of Time Series Analysis*, 15, 183–202.

George, E. I. and McCulloch, R. E. (1993), "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, 88, 881–889.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009), *The Elements of Statistical Learning*, vol. 2, Springer.

Hoegh, A. and Leman, S. (2014), "A spatio-temporal model for assessing winter damage risk to East Coast vineyards," *Journal of Applied Statistics*, 1–12.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999), "Bayesian model averaging: a tutorial," *Statistical Science*, 382–401.

Kalman, R. E. (1960), "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, 82, 35–45.

Liu, J. and West, M. (2001), "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo methods in practice*, Springer, pp. 197–223.

Madigan, D., York, J., and Allard, D. (1995), "Bayesian graphical models for discrete data," *International Statistical Review*, 215–232.

Matérn, B. (1960), "Spatial variation. Stochastic models and their application to some problems in forest surveys and other sampling investigations." *Meddelanden fran statens Skogsforskningsinstitut*, 49.

Niemi, J. (2009), "Bayesian analysis and computational methods for dynamic modeling," Ph.D. thesis, Duke University.

Pitt, M. K. and Shephard, N. (1999), "Filtering via simulation: auxiliary particle filters," *Journal of the American Statistical Association*, 94, pp. 590–599.

Rojas, R. (2009), "AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting," *Freie University, Berlin, Tech. Rep*.

Rolka, H., Burkom, H., Cooper, G. F., Kulldorff, M., Madigan, D., and Wong, W.-K. (2007), "Issues in applied statistics for public health bioterrorism surveillance using multiple data streams: research needs," *Statistics in Medicine*, 26, 1834–1856.

Schapire, R. E. (1990), "The strength of weak learnability," *Machine Learning*, 5, 197–227.

Tobler, W. R. (1970), "A computer movie simulating urban growth in the Detroit region," *Economic Geography*, 234–240.

Venkataramani, K. and Kumar, B. (2006), "Role of statistical dependence between classifier scores in determining the best decision fusion rule for improved biometric verification," in *Multimedia Content Representation, Classification and Security*, eds. Gunsel, B., Jain, A., Tekalp, A., and Sankur, B., Springer Berlin Heidelberg, vol. 4105 of *Lecture Notes in Computer Science*, pp. 489–496.

Waller, L. A., Carlin, B. P., Xia, H., and Gelfand, A. E. (1997), "Hierarchical spatio-temporal mapping of disease rates," *Journal of the American Statistical Association*, 92, pp. 607–617.

West, M. and Harrison, J. (1997), *Bayesian Forecasting and Dynamic Models*, vol. 2, Springer New York.

Wolpert, D. H. (1992), "Stacked generalization," *Neural Networks*, 5, 241 – 259.

Chapter 2

Correlated Model Fusion

ANDREW HOEGH AND SCOTLAND LEMAN
DEPARTMENT OF STATISTICS, VIRGINIA TECH, BLACKSBURG, VA 24061
MANUSCRIPT IN PREPARATION FOR SUBMISSION

Abstract

Model fusion methods, or more generally ensemble methods, are a useful tool for prediction. Combining predictions from a set of models smoothes out biases and reduces variances of predictions from individual models, and hence, the combined predictions typically outperform those from individual models. In many situations individual predictions are arithmetically averaged with equal weights; however, in the presence of correlated models the fusion process is required to account for association between models. Otherwise, the naively averaged predictions will be suboptimal. This chapter describes optimal model fusion principles and illustrates the potential pitfalls of naive fusion in the presence of correlated models for binary data. An efficient algorithm for correlated model fusion is detailed and applied to algorithms mining social media information to predict civil unrest.

Keywords: Ensemble Methods, Gaussian Graphical Models, Model Averaging, Model Selection

2.1 Introduction

In the era of big data, applications for predictive modeling are abundant. For instance, to date the popular website www.Kaggle.com has hosted over 160 data modeling competitions and given away over 2.5 million dollars in prize money. One avenue that has been particularly fruitful for predictive modeling is model fusion, which combines several models to obtain a prediction. In fact, this type of method has been quite successful in data modeling competitions such as the Netflix prize (Bell et al., 2010) and those hosted on Kaggle (Lopez and Matthews, 2015). While in general, model fusion, or model averaging, techniques tend to result in improved overall predictions, many existing methods do not explicitly address model correlation. Failing to account for model correlation can lead to suboptimal predictions, particularly when model association is strong. Therefore, primary considerations for model fusion need to include not only the predictive ability of each individual model, but also the correlation between individual models. In this chapter we consider fusing a fixed set of models producing binary output, that is we condition the fusion process on output from the set of models. We present a fully Bayesian model fusion procedure that learns the association structure between models. This explicit characterization of the association between models is implemented using a Gaussian Graphical Model (GGM) on a latent representation of binary output induced by a multinomial probit model. This framework permits model averaging across graph structures and inferential studies of model association and most probable graphs. An efficient algorithm is presented and applied to simulation studies and a dataset predicting civil unrest.

2.1.1 Overview of Fusion Methods

Many modern prediction problems have vast, heterogenous data sets in which joint processing across unstructured data is implausible, see for example Ramakrishnan et al. (2014). A solution to this scenario is a modular approach to prediction which allows several separate models to be constructed. While this is conceptually attractive, it does require a mechanism for combining or fusing the model output. In other scenarios, multiple models may not be necessary, but can be

beneficial (Breiman, 1996) for improved prediction. Either scenario requires a procedure to discern a consensus decision. In essence, what we denote the fusion process is an ensemble classifier. To provide contrast to a common class of ensemble methods, our fusion process conditions on the model outputs. For instance, in our application various research groups independently develop models which require vast domain knowledge and ultimately these models are fused together. This is different from the scenario in boosting (Schapire, 1990) or a random forest (Breiman, 2001) in which a large number of models can be constructed by the ensemble mechanism itself. Nevertheless, many off the shelf prediction methods, such as those in Kittler et al. (1998), can be applied to scenarios in this chapter - namely classification problems with binary inputs; however, we show that failing to explicitly account for model correlation results in suboptimal predictions.

A simple greedy approach to model fusion is to select predictions from the best model of the batch. However except for extreme scenarios, combining models is always beneficial (Breiman, 1996). A common technique that incorporates all models is majority rule, or simple averaging, which is an equally weighted averaging technique that several ensemble methods incorporate. For instance, the random forest algorithm uses majority rule to combine the set of classification trees. With binary data the majority rule algorithm reduces to a voting algorithm. While this is not a sophisticated technique, it is the optimal decision rule in many classification scenarios and serves as a logical benchmark. One drawback of voting methods is that they do not naturally produce probabilistic predictions from a model based framework. With a large number of models the average of votes could be converted to a probability, but the scenarios in this chapter consist of a small number of models, typically less than 10, leading to probabilities with coarse resolution.

In contrast to arbitrarily selecting the best model or implementing majority rule, an alternative is to learn model weights from the data. Several off-the-shelf classifiers can be used in this capacity including: logistic regression, naive Bayes, neural networks, and Bayesian networks. While these methods are computationally expedient, with the exception of a Bayesian network, they do not explicitly capture between model correlation. The multivariate probit framework we propose is different from a Bayesian network in that we learn the joint distribution directly rather than a product of conditional distributions. Furthermore, a Bayesian network requires knowing (or learn-

ing) the association structure between models in order to decompose the joint distribution into the conditional distributions. Our method, using Gaussian graphical models to select graph structures, provides a mechanism to incorporate uncertainty in the association structure and the ability to average across different association structures.

Another related category of the statistical literature is that of model selection and averaging. In particular, Bayesian model averaging (Hoeting et al., 1999) implemented by a variable selection algorithm is a powerful method that tends to down weight similar models; however, correlation is not handled in an explicit manner. Similarly, ensemble Bayesian Model Averaging (Raftery et al., 2005; Sloughter et al., 2010) and ensemble copula coupling (Schefzik et al., 2013) are used in weather forecasting to post process and combine output from deterministic computer ensembles. Other methods such as LASSO (Tibshirani, 1996) and Grouped LASSO (Yuan and Lin, 2006) are often used for variable selection under correlated variables. However, again these methods do not explicitly handle correlation between variables or models.

2.1.2 Chapter Overview

The remainder of this chapter is as follows. Section 2.2 details the data structures considered in this paper and presents decision theory for optimal predictions. Section 2.3 details our fusion method for correlated binary data to produce binary and probability predictions. Section 2.4 contains simulations studies showing the algorithms ability to recover the model association structure and issue predictions. Section 2.5 implements our methods to predictions for civil unrest and Section 2.6 concludes with a discussion. Section 2.7, the appendix, contains pseudocode and the full conditionals for the MCMC algorithm for implementing our algorithm.

2.2 Data Structure and Decision Theory

In this chapter we consider the problem of issuing a consensus binary prediction by integrating binary predictions from a collection of models. In some cases the goal is a classification decision, that is a binary response, and in other cases we issue a probability of an event occurring. The motivating problem is predicting civil unrest. A set of algorithms mine social media information in order to issue binary predictions as to whether a protest will occur at a given locale. In this case the underlying algorithms that are used to issue predictions are a secondary concern. Rather, the primary focus is on the procedures and algorithms for combing models.

To evaluate predictions from different frameworks, we introduce the loss functions necessary for this evaluation. A loss function, $L(\theta, \delta(\boldsymbol{x}))$ evaluates an estimator $\delta(\boldsymbol{x})$ by penalizing values differing from the observed response θ. We present two loss functions, one associated with binary predictions and another associated with probability predictions. For binary predictions, we use the general loss function shown in Table 2.1.

		Predicted	
		$\delta(\boldsymbol{x}) = 1$	$\delta(\boldsymbol{x}) = 0$
Actual	$\theta = 1$	c_{11}	c_{10}
	$\theta = 0$	c_{01}	c_{00}

Table 2.1: Loss Function Matrix: $L(\theta, \delta(\boldsymbol{x}))$

For most applications $c_{11} = c_{00} = 0$, as there is no penalty for correct predictions. Then when $c_{01} = c_{10} = 1$ the resulting loss function is known as zero-one loss. Minimizing the zero-one loss function is analogous to minimizing classification error. For cases when probabilistic predictions are used the following loss function is utilized:

$$L(\theta, \delta(\boldsymbol{x})) = \theta \log(\delta(\boldsymbol{x})) + (1 - \theta) \log(1 - \delta(\boldsymbol{x})), \tag{2.1}$$

where $\theta \in \{0, 1\}$ and $\delta(\boldsymbol{x}) \in (0, 1)$. This loss function is the log-likelihood of a Bernoulli random

variable and is often called the binomial deviance. This loss function sharply penalizes predictions that are near 0 or 1 when they are wrong. It can be shown that this is a proper scoring function (Gneiting and Raftery, 2007), meaning the true probabilities are optimal in terms of minimizing the expected loss. While a loss function defines the penalty for an instance of θ and an associated $\delta(\boldsymbol{x})$, optimality of an estimator is defined as the estimator that minimizes the expected loss, or risk:

$$R(\theta, \delta) = E[L(\theta, \delta(\boldsymbol{x}))].$$

Risk provides a way to compare two estimators $\delta_1(\boldsymbol{x})$ and $\delta_2(\boldsymbol{x})$ by averaging the loss with respect to θ.

2.2.1 Optimal Solutions

As a means for characterizing the uncertainty in θ, a common approach is to use a Bayesian framework to compute Bayes risk:

$$BR(\delta, \pi) = \int_{\Theta} R(\theta, \delta)\pi(\theta)d\theta,$$

with the goal of identifying the estimator with the smallest Bayes risk. This estimator is known as Bayes rule, for a given prior π and is often denoted δ^{π}. Finding Bayes rule has some computational advantages as Bayes risk factorizes nicely:

$$\begin{aligned}
\int_{\Theta} R(\theta, \delta)\pi(\theta)d\theta &= \int_{\Theta} \left[\int_{\mathcal{X}} L(\theta, \delta(\boldsymbol{x}))f(\boldsymbol{x}|\theta)d\boldsymbol{x} \right] \pi(\theta)d\theta \\
&= \int_{\Theta} \left[\int_{\mathcal{X}} L(\theta, \delta(\boldsymbol{x}))\pi(\theta|\boldsymbol{x})m(\boldsymbol{x})d\boldsymbol{x} \right] \pi(\theta)d\theta \\
&= \int_{\mathcal{X}} \left[\int_{\Theta} L(\theta, \delta(\boldsymbol{x}))\pi(\theta|\boldsymbol{x})\pi(\theta)d\theta \right] m(\boldsymbol{x})d\boldsymbol{x}
\end{aligned}$$

where the value contained in the brackets in Equation 2.2 is the posterior expected loss. Using this factorization, the estimator that minimizes the posterior expected loss also minimizes the Bayes risk. This framework provides a mechanism for finding optimal estimators, if not analytically, at least numerically.

We present a theoretical framework for optimal model fusion of binary predictions for classification problems. Given a set of binary values from models: $\boldsymbol{m} = (m_1, m_2, ..., m_p)$, a decision rule $\delta(\boldsymbol{m})$ is constructed to map $\boldsymbol{m} \rightarrow \{0, 1\}$. To make an optimal decision for the loss function specified in Table 2.1, $p_y = Pr[Y = 1|\boldsymbol{m}]$ is needed, then $\delta_{opt}(\boldsymbol{m})$ under the loss function specified in Table 2.1 is:

$$\delta_{opt}(\boldsymbol{m}) = \begin{cases} 1 & : p_y(c_{11} + c_{01}) \geq (1 - p_y)(c_{10} + c_{00}) \\ 0 & : p_y(c_{11} + c_{01}) < (1 - p_y)(c_{10} + c_{00}) \end{cases}. \tag{2.2}$$

The challenging aspect of this result is to compute p_y. When the models are conditionally independent given y, p_y can be factorized in a trivial manner:

$$p_y = P[Y = 1|m_1, m_2, ..., m_p] = \frac{P[m_1|Y = 1]P[m_2|Y = 1]...P[m_p|Y = 1]P[Y = 1]}{P[m_1, m_2, ..., m_p]}; \tag{2.3}$$

however, under the scenario where models are correlated, the factorization in Equation (2.3) does not apply and the full joint distribution of the models is required for computation. We propose using a multivariate probit model to handle correlation between the models and compute $P[m_1, m_2, ..., m_p|Y = 1]$ and $P[m_1, m_2, ..., m_p|Y = 0]$. Then, an optimal decision for a given set of model inputs $\boldsymbol{m} = (m_1, m_2, ..., m_p)$, can be calculated as

$$p_y = P[Y = 1|\boldsymbol{m}] = \frac{P[\boldsymbol{m}|Y = 1]}{P[\boldsymbol{m}|Y = 1] + P[\boldsymbol{m}|Y = 0]}. \tag{2.4}$$

Thus the optimal decision, given $\boldsymbol{m}$, follows from Equation (2.2). Similarly in the case of the loss function in Equation 2.1, p_y needs to be computed. As this loss function is a proper scoring rule (Gneiting and Raftery, 2007), p_y minimizes the risk.

2.3 Model Fusion Framework

The model fusion process can be formulated as a function that maps $\boldsymbol{m} \rightarrow y$, where $\boldsymbol{m} = \{m_1, ..., m_p\}$ is a vector of predictions from the set of p models and $y \in \{0, 1\}$ for the binary

case and $y \in (0,1)$ for probabilities. This section contains four parts: a description of the multivariate probit model, an overview of Gaussian graphical model theory, an outline of the prior specifications, and details regarding the algorithmic implementations of our novel method.

2.3.1 Multivariate Probit Model

To account for model correlation, we appeal to the multivariate probit model presented in Chib and Greenberg (1998). With this data augmentation procedure latent variables $\boldsymbol{Z_i}$ are distributed as $N(\boldsymbol{Z_i}; \mu, R)$, such that:

$$m_i = \begin{cases} 1 & : z_{ij} \geq 0 \\ 0 & : z_{ij} < 0 \end{cases},$$

where Z_i is the latent vector of responses for the i^{th} trial and z_{ij} and m_{ij} are the latent and binary representation of the i^{th} trial and j^{th} model, respectively. The covariance matrix, R, is constrained to be a correlation matrix for identifiability. With this latent formulation, we leverage Gaussian Graphical Model (GGM) theory for model association. GGMs are used to visualize and encode association structures. Given the latent normal specification, two models are conditionally independent if the corresponding element in the precision matrix (R^{-1}) is zero. Additional details including sampling techniques for GGMs are provided in the next subsection. The likelihood of the latent variables can be formulated as:

$$P[\mathbf{Z}|Y = k, \boldsymbol{\mu}_k, \mathbf{R}_k, G_k] = \prod_i (2\pi)^{-p/2} |\mathbf{R}_k|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{Z_i} - \boldsymbol{\mu}_k)' \boldsymbol{R}_k^{-1} (\mathbf{Z_i} - \boldsymbol{\mu}_k)), \qquad (2.5)$$

where the mean, $\boldsymbol{\mu}_k$, correlation, R_k, and graph structure, G_k, vary whether $Y = \{0, 1\}$. Then given the likelihood of the latent variables, $P[\boldsymbol{Z}|Y = k, \boldsymbol{\mu}_k, \boldsymbol{R}_k, G_k]$, the binary representation $P[\boldsymbol{M}|Y = k, \boldsymbol{\mu}_k, \boldsymbol{R}_k, G_k]$ is obtained by integrating out the latent variables, which turns to to require integrating the area in each of the 2^p orthants, as follows:

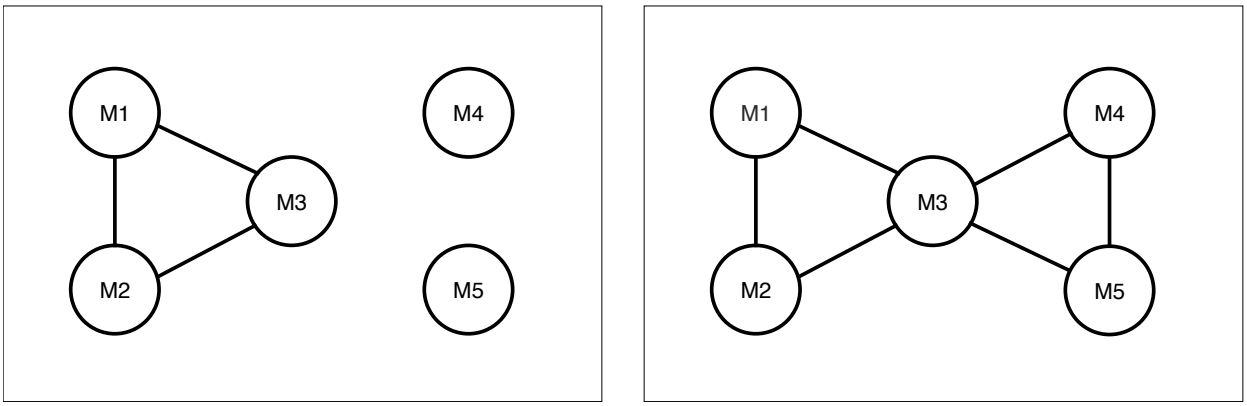
$$P[\boldsymbol{M}|Y = k, \boldsymbol{\mu}_k, \boldsymbol{R}_k, G_k] \;=\; \int_{Z_*} P[\mathbf{Z}|Y = k, \boldsymbol{\mu}_k, \mathbf{R}_k, G_k] d\boldsymbol{Z}, \qquad (2.6)$$

where $\boldsymbol{Z}_*$ denotes the quadrants such that $z_j > 0$ if $M_j = 1$ and $z_j < 0$ if $M_j = 0$. In high dimensions this integration would be computationally challenging. However, the model fusion

frameworks that we have considered are of a small dimension. The integration of Equation 2.6 can be computed using an accept-reject sampling method from Robert and Casella (2005).

2.3.2 Gaussian Graphical Models and Junction Trees

A graph can be characterized by a set of vertices, or nodes, denoted as V and edges, E, connecting the vertices. In this work we focus exclusively on undirected graph structures where all edges connecting nodes are bi-directional. Two nodes that are not connected by an edge are conditionally independent given the other nodes contained in the graph. For instance, consider Figure 2.5a, where each node denotes a model. In this illustration Model 3 (M3) would be conditionally



(a) Graph 1 (b) Graph 2

Figure 2.1: Graph illustrating model association

independent of Model 4 (M4). However, in Figure 2.5b M3 and M4 would not be conditionally independent, but M2 and M4 would be. A subgraph is a subset of the vertices and the associated edges. A subgraph is complete if there is an edge connecting each node in the subgraph. A complete subgraph is known as a clique. For example, in Figure 2.5b M1, M2, and M3 would constitute a clique and M3, M4, and M5 another. A set S is called a separator of subgraphs A and B if the path from A to B must go through S. The collection of subgraphs, including the separator set, form a decomposition of V if $V = A \cup B, S = A \cap B$, where S is complete and separates A

from B. Referring to Figure 2.5b, the separator $S =$ M3, which separates subgraph A consisting of M1, M2, and M3 from that of subgraph B consisting of M3, M4, and M5. A subgraph that cannot be further decomposed is known as a prime connector. The graph is called decomposable if every prime component is complete. For additional details readers are referred to Lauritzen (1996).

GGM introduce a graph structure based on the elements of the inverse covariance matrix, or precision matrix, from a multivariate normal distribution. Nonzero elements in the precision matrix correspond to edges in the graph, whereas zeros indicate conditional independence and consequently that an edge does not exist between the two nodes. In this framework, we use GGM on the latent representation of the binary data as shown in Equation 2.5. Then the density of Z can be factorized as

$$p(\boldsymbol{Z}|R, G) = \frac{\Pi_{p\in\mathcal{P}}\ p(\boldsymbol{Z}_p|R_p)}{\Pi_{s\in\mathcal{S}}\ p(\boldsymbol{Z}_s|R_s)}, \tag{2.7}$$

where p and s correspond to prime components and separators, respectively.

For sampling graph structures it is necessary to provide a unique representation of the graph. This is accomplished through the concept of perfect ordering and the use of a junction tree. A perfect ordering of prime components and separators $(p_1, s_2, p_2, s_3, ...p_k)$ satisfies the running intersection property (Lauritzen, 1996), where $s_i = p_i \cap H_{i-1} \subset p_j$ and $H_{i-1} = \cup_{j=1}^{i-1}$, if there exists a $j < i$. A junction tree is useful for representing prime components with a perfect ordering. A junction tree representation of the graph in Figure 2.5b would be: $p_1 = \{1, 2, 3\}, s_2 = \{3\}, p_2 = \{3, 4, 5\}$. Junction trees are useful for efficient sampling of GGMs. For additional details on sampling procedures for GGMs see Carvalho et al. (2007).

2.3.3 Prior Specification

One of the computational challenges with the multivariate probit model is efficient sampling of the covariance matrix of the latent variables. Due to identifiability reasons the covariance matrix is constrained to be a correlation matrix. Efficient sampling of matrices is a difficult problem and typically conjugate priors are used to facilitate efficient sampling. An Inverse Wishart (IW) prior

is conjugate for a covariance matrix, under Gaussian data, where

$$\boldsymbol{\Sigma} \sim IW(\nu, \boldsymbol{\Omega}) \propto |\frac{\boldsymbol{\Omega}}{2}|^{(\frac{\nu+M-1}{2})} |\boldsymbol{\Sigma}|^{(\frac{-\nu+2M}{2})} \exp(-\frac{1}{2} tr[\boldsymbol{\Sigma}^{-1}\boldsymbol{\Omega}]).$$

Furthermore, a Hyper-Inverse Wishart (HIW) distribution is conjugate for a covariance matrix, under Gaussian data and a specified GGM, where

$$\boldsymbol{\Sigma} \sim HIW(\nu, \boldsymbol{\Omega}) = \frac{\Pi_{p \in \mathcal{P}} \; IW(\boldsymbol{\Sigma}^p; \nu, \boldsymbol{\Omega}^p)}{\Pi_{s \in \mathcal{S}} \; IW(\boldsymbol{\Sigma}^s; \nu, \boldsymbol{\Omega}^s)}.$$

Unfortunately these distributions are not conjugate priors for correlation matrices. Rather, parameter expansion and data augmentation (PXDA) is used to transform the correlation matrix to a covariance matrix for sampling. The sampled covariance matrix is then mapped back to a correlation matrix. To implement PXDA on structured correlation matrices we use procedures detailed in Talhouk et al. (2012), the prior on the correlation matrix follows as:

$$\pi(R) = \frac{\Pi_{p \in \mathcal{P}} \; \pi(R^p)}{\Pi_{s \in \mathcal{S}} \; \pi(R^s)},$$

where

$$\pi(R^j) \propto |R^j|^{\frac{|j|(|j|-1)}{2}-1} \left(\Pi_{i \in j} |R^j_{ii}|\right)^{-\frac{(|j|+1)}{2}}.$$

The matrix R^j_{ii} is the i^{th} principal submatrix of R^j. Finally placing a conjugate prior, flat in this case, on $\boldsymbol{\mu}$ enables the use of PXDA for sampling $\boldsymbol{\mu}$ and R, which is described in more detail in the next subsection. A uniform prior is placed on the graph structure G and the prior for $Y \sim Beta(a, b)$.

2.3.4 Posterior Distributions and Computation

The purpose of using the multivariate probit model along with GGMs is to compute the posterior distribution for p_y, which follows as:

$$p_y = P[Y = 1|\boldsymbol{M}] = \int \frac{P[\boldsymbol{M}|Y = 1, \boldsymbol{\mu}_1, \boldsymbol{R}_1]P[Y = 1]}{P[\boldsymbol{M}|Y = 1, \boldsymbol{\mu}_1, \boldsymbol{R}_1]P[Y = 1] + P[\boldsymbol{M}|Y = 0, \boldsymbol{\mu}_0, \boldsymbol{R}_0]P[Y = 0]} p[\boldsymbol{\Theta}] d\boldsymbol{\Theta}, \tag{2.8}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{R}_1, \boldsymbol{\mu}_1, G_1, \boldsymbol{R}_0, \boldsymbol{\mu}_0, G_0\}$. In essence, we have developed a fully Bayesian discriminate analysis akin to Quadratic Discriminate Analysis (QDA) for a binary response *and* binary inputs.

Pseudocode for the implementation of our algorithm is presented next. A simplified version of this algorithm, excluding sampling the Gaussian graphical model, is presented in Hoegh et al. (2015). Steps (1) - (3) draw heavily on the procedure detailed in Talhouk et al. (2012). Comprehensive derivations of full conditionals and Metropolis with Gibbs sampling procedures are described in the appendix in Section 2.7.

Estimation is conducted using an MCMC procedure as follows:

1. Sample $(\boldsymbol{Z}_1|E = 1, \boldsymbol{\mu}_1, \boldsymbol{R}_1)$ and $(\boldsymbol{Z}_0|E = 0, \boldsymbol{\mu}_0, \boldsymbol{R}_0) \sim N(\boldsymbol{\mu}_i, \boldsymbol{R}_i)$. For computational efficiency the multivariate densities can be decomposed into univariate normal densities, then sampling proceeds using importance sampling as in Robert (1995). The use of standard accept-reject samplers or importance sampling with a multivariate (or univariate) normal distribution is unfeasibly slow, due to truncated regions having extremely low probabilities.

2. Sample $(\boldsymbol{\mu}_1, \boldsymbol{R}_1|\boldsymbol{Z}_1, G_1)$ and $(\boldsymbol{\mu}_0, \boldsymbol{R}_0|\boldsymbol{Z}_0, G_0)$ using PXDA as in Talhouk et al. (2012) and sample $(G_0|\boldsymbol{\mu}_0, \boldsymbol{R}_0, \boldsymbol{Z}_0)$ and $(G_1|\boldsymbol{\mu}_1, \boldsymbol{R}_1, \boldsymbol{Z}_1)$ using a Metropolis-Hastings procedure.

   (a) Sample $(d_{ii}|\boldsymbol{R}) \sim IG((J+1)/2, r^{ii}/2)$, where J is the number of models and r^{ii} are the diagonal elements of $\boldsymbol{R}^{-1}$. Then let $\boldsymbol{D}$ be a matrix with diagonal elements d_{ii} and compute $\boldsymbol{W} = \boldsymbol{Z}\boldsymbol{D}$. Then $\boldsymbol{\Sigma} = \boldsymbol{D}\boldsymbol{R}\boldsymbol{D}$ and $\boldsymbol{\gamma} = \boldsymbol{\mu}\boldsymbol{D}$.

   (b) Sample $(\boldsymbol{\Sigma}|\boldsymbol{W})$ from $HIW\left(2+n, \boldsymbol{W}'\boldsymbol{W} + \boldsymbol{I} - \boldsymbol{W}' J_n \boldsymbol{W}/(n+1))\right)$, where J_n is a $n \times n$ matrix consisting of 1s, using the junction tree decomposition shown in Carvalho et al. (2007).

   (c) Sample $(\boldsymbol{\gamma}|\boldsymbol{W}, \boldsymbol{\Sigma}) \sim N\left(\mathbf{1}_n' \boldsymbol{W}/(n+1), \Sigma/(n+1))\right)$.

   (d) Sample $(G|W)$ using a Metropolis-Hastings proposal, where for the proposal two vertices are randomly selected and the edge between those vertices is added or removed depending on whether the edge currently exists.

   (e) Compute $\boldsymbol{R} = \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}$ and $\boldsymbol{\mu} = \boldsymbol{\gamma}\boldsymbol{Q}$, where $\boldsymbol{Q}$ is a diagonal matrix with diagonal elements $q_{ii} = \sigma_{ii}^{-1/2}$, where $\sigma_{ii}^{-1/2}$ are the diagonal elements of the precision matrix Σ^{-1}.

3. Compute $P[\boldsymbol{M}|Y=1,\boldsymbol{\mu}_1,\boldsymbol{R}_1,G_1]$ and $P[\boldsymbol{M}|Y=0,\boldsymbol{\mu}_0,\boldsymbol{R}_0,G_0]$ as in (2.6) for each possible set of model inputs $\boldsymbol{M}\in\mathcal{M}$.

4. Sample $P[Y=1]\sim Beta(a,b)$

5. $P[E=1|\boldsymbol{M},\boldsymbol{\mu}_1,\boldsymbol{R}_1,G_1,\boldsymbol{\mu}_0,\boldsymbol{R}_0,G_0]$ is computed via (2.8), where each iteration of the MCMC sampler integrates over $\{\boldsymbol{\mu}_1,\boldsymbol{R}_1,G_1,\boldsymbol{\mu}_0,\boldsymbol{R}_0,G_0\}$ to obtain $P[Y=1|\boldsymbol{M}]$ for $\boldsymbol{M}\in\mathcal{M}$.

2.4 Simulation Study

In this section, we present a scenario involving correlated models. In particular, this set of simulation studies contains scenarios where the magnitude of model correlation varies and we examine the performance of our Bayesian algorithm to recover the model structure and issue predictions. Consider the five models depicted in the GGM shown in Figure 2.5a, where model association is encoded by elements in the precision matrix Σ^{-1}. In this figure Models 1, 2, and 3 constitute a clique of models, which are conditionally independent of Model 4 and Model 5. However, Models 1, 2, and 3 themselves are correlated. The degree of correlation is varied in our simulations.

Data is generated using the latent normal representation of binary data popularized by Albert and Chib (1993),

$$\boldsymbol{Z}|\boldsymbol{Y}\sim N(\boldsymbol{\mu_y},\Sigma). \tag{2.9}$$

Then m_i, the ith element of the binary response is $\mathbf{1}(z_i>0)$, where $\mathbf{1}(.)$ is an indicator function. For this scenario model parameters are set as follows:

$$\mu_1=\begin{pmatrix}0.524\\0.524\\0.524\\0.842\\0.842\end{pmatrix},\mu_0=\begin{pmatrix}-0.524\\-0.524\\-0.524\\-0.842\\-0.842\end{pmatrix},\text{and}\quad \Sigma_1=\Sigma_2=\begin{pmatrix}1&\rho&\rho&0&0\\\rho&1&\rho&0&0\\\rho&\rho&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix}.$$

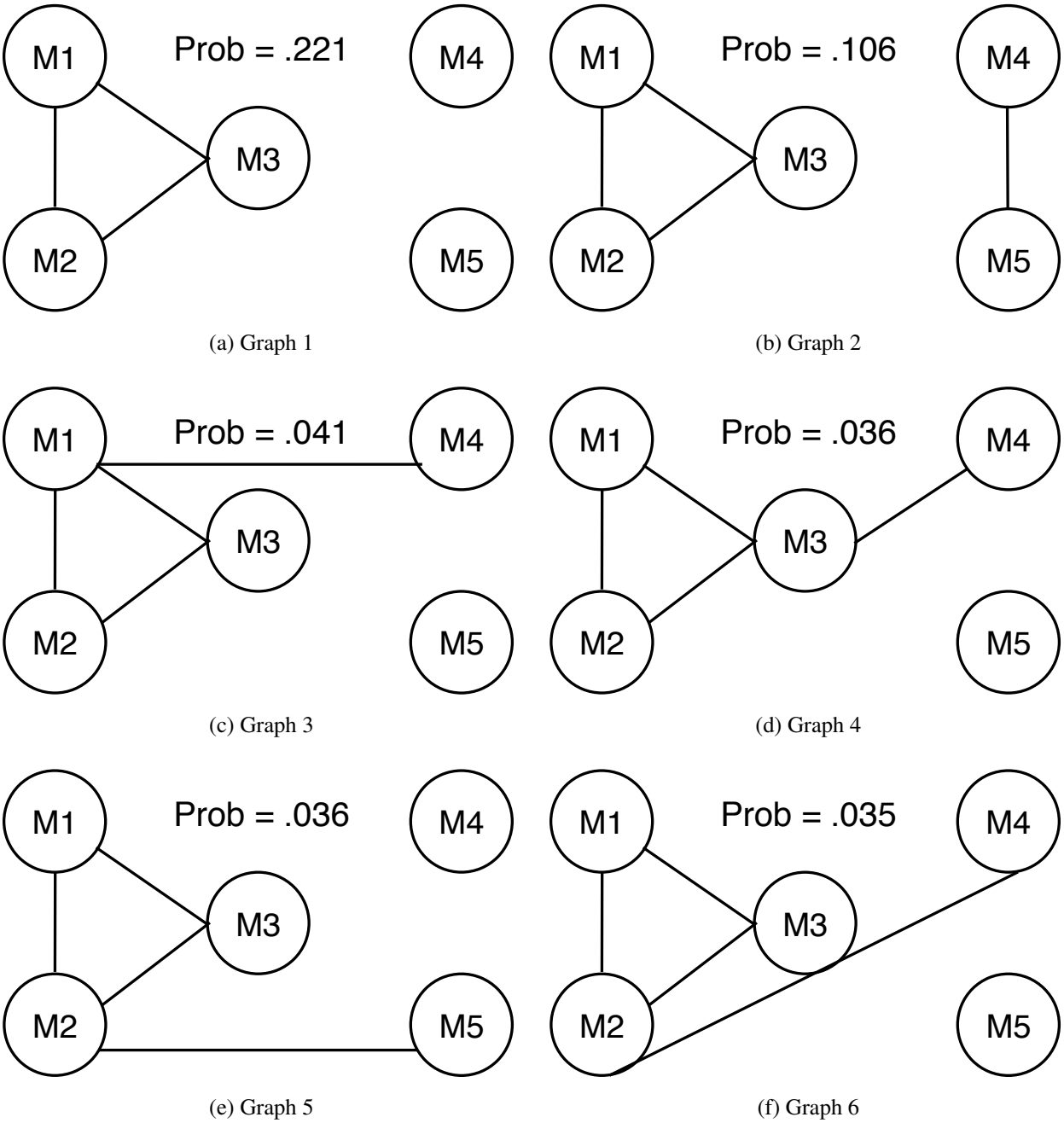
The correlation, ρ, varies from a moderate amount, 0.3, to extremely high correlation, 1.0, across the simulations and $\Phi^{-1}(0.524) = .7, \Phi^{-1}(0.842) = .8$ where $\Phi(x)^{-1}$ is the inverse cumulative distribution function for a standard normal random variable. Under this scenario, in addition to learning the association structure of the models the strength of the individual models plays an important role as some models make correct predictions with greater probabilities than the other models.

Graph Selection

Initially we consider the ability of our algorithm to recover the most probable graph structures. For ease of displaying the results, we choose a single value of ρ, the correlation between the models in a clique. In particular we set $\rho = 0.9$ in this simulation. In some scenarios the graph structure may be the goal of the analysis, but in other scenarios the graph structure is a necessary component in producing accurate predictions. Our algorithm allows different graph structures for cases where $Y = 1$ or $Y = 0$, but in this simulation the graph structures are set to be the same.

The most probable graphs are shown in Figure 2.2. We recover the actual graph structure as the most probable graph; furthermore, the next most probable graphs all consist of one additional edge from the true structure.

In addition to most probable graph structures, our model also allows us to examine the marginal probability of inclusion for each potential edge in the graph. Figure 2.3 contains the marginal probabilities of inclusion for each edge. Using our algorithm, the models that are simulated from the same clique have extremely high posterior probability of edge inclusion, generally greater than 0.95. Similarly, models that are not connected have very low posterior probability of edge inclusions, generally less than $.20$. Again in this case we compute inclusion probabilities conditional on $Y = 1$ and $Y = 0$; however, given that the true structure is the same for both in the simulation we choose one without loss of generality.



(a) Graph 1

(b) Graph 2

(c) Graph 3

(d) Graph 4

(e) Graph 5

(f) Graph 6

Figure 2.2: Graph illustrating most probable model association

n=2000

Prediction

In addition to learning the association structure between models, fusion also enhances prediction. In Figure 2.4, the risk profiles corresponding to classification error and binomial deviance are

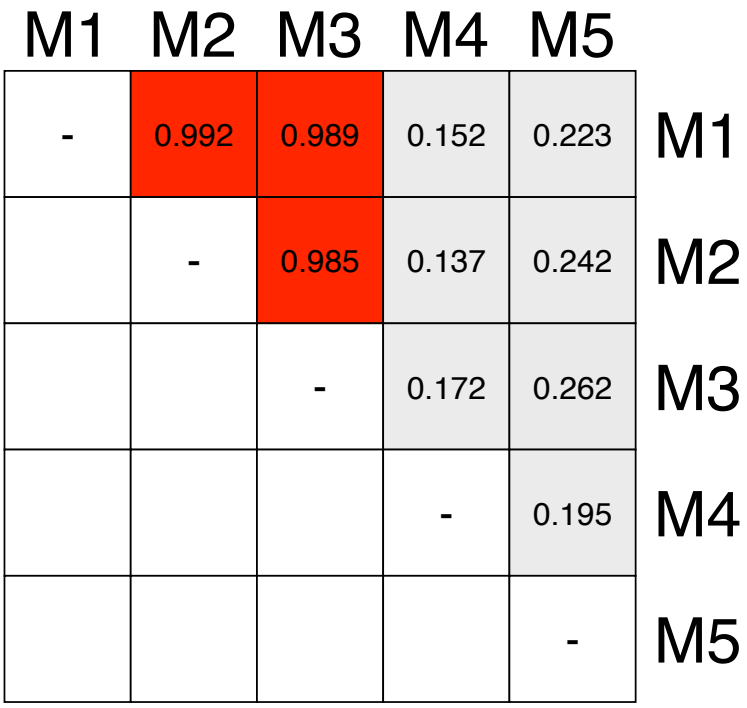
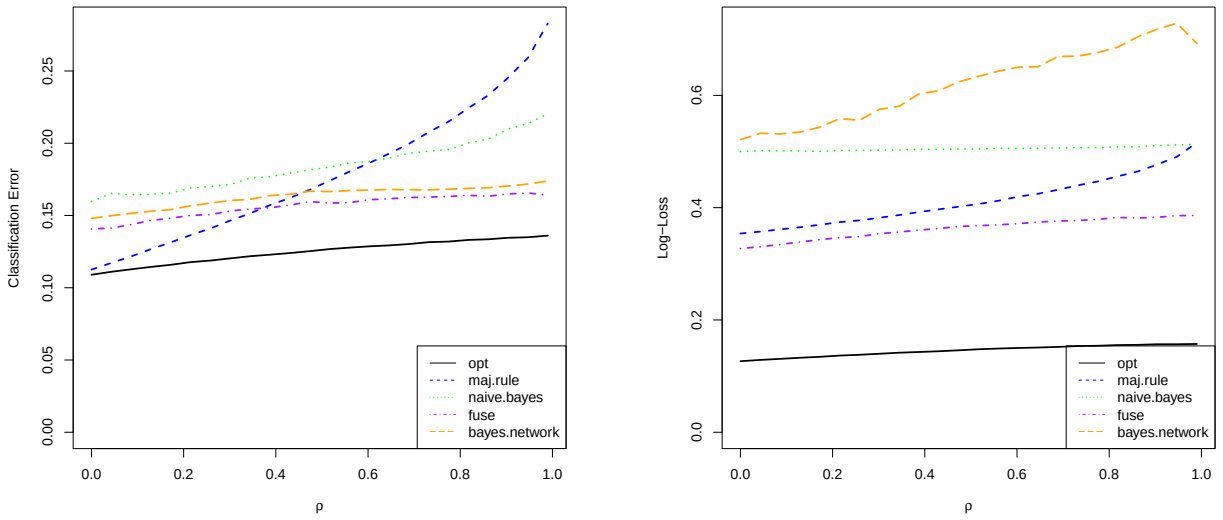


Figure 2.3: Marginal inclusion probabilities for each edge.

shown for our multivariate probit fusion model along with competing methods. In particular, we make comparisons with a majority rule algorithm, naive Bayes, and a Bayesian network. For a historical overview of naive Bayes see Lewis (1998). The Bayesian network structure is learned using the hill climbing algorithm detailed in Scutari (2010). The graphics also show the optimal decision. For this simulation, a fairly small sample size, of 50 observations, is used along with the same model specified for the simulation on the graph structure.

Given the smaller dataset, it is not surprising that all of the methods differ from the optimal decision. Nevertheless, the multivariate probit model that we propose is the best for all correlation structures for Log-Loss function detailed in Equation 2.1. The multivariate probit model with the induced graph structure is also superior for classification error with higher correlation structures. The majority rule works well (in this particular case) when there is little correlation.



(a) Classification Error

(b) Log Loss

Figure 2.4: Risk for a given training sample size. Green = naive Bayes, purple = multivariate probit, orange = Bayesian network, blue = majority rule, and black is optimal

2.5 Application: Modeling Civil Unrest

The methodology and algorithmic implementation detailed herein was developed to fuse a set of models designed to predict protests as detailed in Ramakrishnan et al. (2014). The models, or predictive algorithms, are developed independently (by collaborators), but likely have some association structure given similar or overlapping data sources are used for construction. Hence, we apply our multivariate probit model to learn the association structure and make fused predictions.

For this analysis, we have civil unrest data from six capital cities in Central and South America: Bogotá, Colombia; Buenos Aires, Argentina; Caracas, Venezuela; Mexico City, Mexico; Montevideo, Uruguay; and San Salvador, El Salvador. The data contains a binary indicator on whether a protest was recorded in that city on a given day as well as binary predictions from a set of five predictive algorithms. The data spans nine months from July 2013 through March 2014. We are interested in learning the association structure between the algorithms in each of the cities as well

identifying cities with large numbers of predictions or where the models are strong or poor.

2.5.1 Overview of Civil Unrest Algorithms

The five predictive models that we will be fusing are the Historical Protest (HP), Dynamic Query Extraction (DQE), Planned Protest (PP), Spatial Scan (SS), and Keyword Volume (KV). Here we will explain the basics of each model. Additional details on each model are provided in Section 3.2.1, where the models are used in a different framework. Some of these models have the capability of issuing multiple predictions for a given location and date, but for this analysis we consider binary outcomes that is whether or not one or more protest will occur.

The HP model is a backward looking model that issue predictions corresponding to event-location pairs that occur with high frequency. The DQE model uses keywords from twitter to detect evolving terms that relate to protests. The keywords are learned dynamically and correspond to current protests. The PP model mines information from news, blogs, and other publicly available social media forums in order to capture upcoming protests. The SS model track clusters of tweets over time. Clusters that grow in time indicate future protest events. The KV using keywords from twitter to construct a logistic regression model, with a LASSO penalty (Tibshirani, 1996) for predictions.

2.5.2 Estimated Model Parameters

An independent implementation of our modeling framework is fit for each city. The results will be presented it two ways. First, we will assess the association structure of the models along with mean parameters from our multivariate probit model using all of the data in a retrospective analysis. Then we will use half of the data to learn predictive probabilities of an event and use these probabilities to forecast upcoming protests. The predictions will be compared using both the log loss and classification error frameworks described earlier.

Table 2.2 and Table 2.3 summarize the properties of the predictive models as well as the frequency

of protests in the capital city of each country. With an individual model there are two factors

Table 2.2: Frequency of civil unrest events and models issuing alerts

Country	$P(HP=1)$	$P(DQE=1)$	$P(PP=1)$	$P(SS=1)$	$P(KV=1)$	$P(E=1)$
Argentina	0.58	0.47	0.11	0.19	0.73	0.57
Colombia	0.19	0.52	0.14	0.18	0.31	0.34
El Salvador	0.42	0.47	0.00	0.09	0.48	0.40
Mexico	0.69	0.82	0.27	0.01	0.63	0.71
Uruguay	0.47	0.51	0.02	0.04	0.54	0.39
Venezuela	0.64	0.81	0.38	0.33	0.38	0.68

that are important: 1) how often it issues an alert and, 2) the probability of each response (yes or no) from a model given when an event happens and does not happen. Table 2.2 corresponds to the first factor, by displaying the frequency of models issuing an alert for a protest and the frequency of an event occurring $(E = 1)$. The second factor is shown in Table 2.3, which displays

Table 2.3: Posterior Means of Model Means from Multivariate Probit Fusion

Country	$\mu_{0(HP)}$	$\mu_{0(DQE)}$	$\mu_{0(PP)}$	$\mu_{0(SS)}$	$\mu_{0(KV)}$	$\mu_{1(HP)}$	$\mu_{1(DQE)}$	$\mu_{1(PP)}$	$\mu_{0(SS)}$	$\mu_{1(KV)}$
Argentina	-0.18	-0.26	-1.40	-0.89	0.56	0.51	0.07	-1.02	-0.83	0.62
Colombia	-0.86	0.04	-1.27	-0.88	-0.52	-0.83	0.07	-0.73	-0.91	0.41
El Salvador	-0.37	0.02	-2.50	-1.19	-0.03	0.07	-0.17	-2.36	-1.45	-0.06
Mexico	0.35	0.75	-0.74	-1.93	0.32	0.55	0.98	-0.54	-2.58	0.33
Uruguay	-0.24	-0.05	-1.78	-1.65	0.09	0.18	0.16	-2.08	-1.68	0.08
Venezuela	0.16	0.61	-0.71	-0.51	-0.48	0.45	0.96	-0.12	-0.39	-0.20

the posterior means of the $\boldsymbol{\mu}$ terms from our multivariate probit framework. Making a collective prediction requires jointly evaluating all of the models to incorporate the association structure; nevertheless, these tables provide marginal properties of each model. From Table 2.2 it is apparent that some of the models issue fewer alerts. An ideal model would issue the alerts at the same

frequency as the occurrence of protest events. Furthermore, this model would also have strong predictive ability, which would be represented by a large negative term for $\boldsymbol{\mu_0}$ corresponding to a low probability of issuing an alert when an event does not occur along with a large positive term for $\boldsymbol{\mu_1}$ corresponding to a high probability of issuing an alert when an event does happen. A model with little or no predictive ability would have $\boldsymbol{\mu_0}$ approximately equal to $\boldsymbol{\mu_1}$. Looking at the table we see that, individually certain models tend to be better than others and protests may be more difficult to predict in certain countries.

Next we consider the association structure of the predictive models. Table 2.4 gives posterior probabilities for the number of edges on the graphical structures for each country when an event occurs and when an event does not occur. While there are more possible graphs with four, five, or six edges, these totals suggest well connected graphs. In this case there are a total of 1024 possible

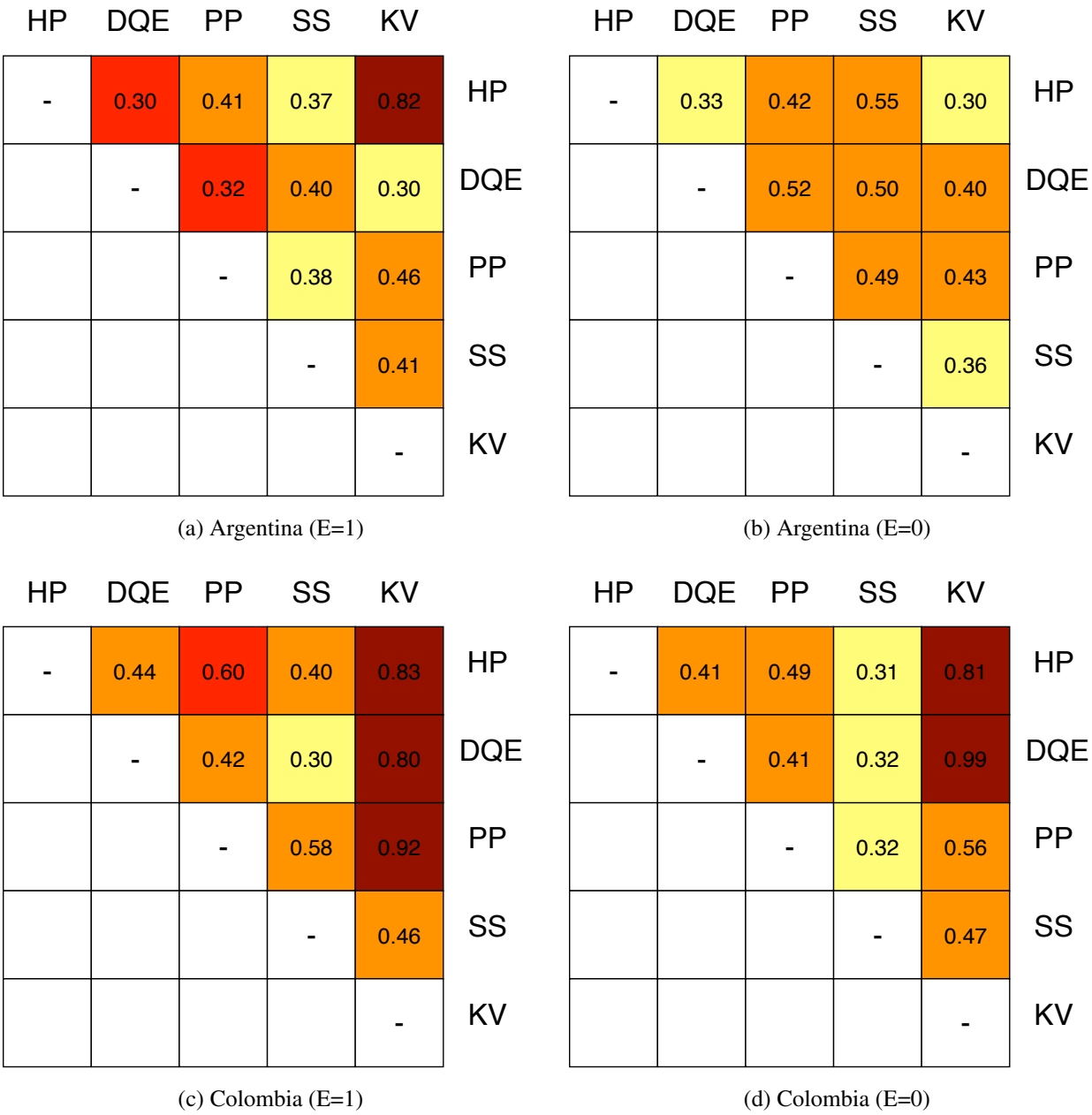
Table 2.4: Posterior Probability of Graph Edges

# Edges	Arg_1	Arg_0	Col_1	Col_0	El_1	El_0	Mex_1	Mex_0	Ur_1	Ur_0	Ven_1	Ven_0
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.02	0.02	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.01	0.00	0.00
2	0.09	0.08	0.00	0.02	0.01	0.04	0.04	0.05	0.01	0.03	0.01	0.02
3	0.22	0.20	0.03	0.10	0.07	0.13	0.16	0.15	0.05	0.13	0.07	0.11
4	0.30	0.29	0.16	0.24	0.18	0.27	0.30	0.26	0.14	0.27	0.20	0.24
5	0.20	0.21	0.25	0.25	0.24	0.25	0.23	0.23	0.21	0.25	0.30	0.25
6	0.11	0.12	0.26	0.22	0.23	0.18	0.15	0.16	0.25	0.17	0.26	0.21
7	0.05	0.06	0.22	0.13	0.18	0.10	0.08	0.10	0.21	0.10	0.14	0.11
8	0.01	0.02	0.07	0.03	0.05	0.03	0.03	0.03	0.10	0.03	0.03	0.03
9	0.00	0.00	0.02	0.01	0.02	0.00	0.01	0.01	0.04	0.01	0.01	0.01
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

graphs. The total number of graphs with each i of edges can be computed as $\binom{10}{i}$, where 10 are the unique number of non-diagonal elements in the the precision matrix.

We also display posterior predictive probabilities of two model sharing an edge, which can be seen

in Figure 2.5. These values are not correlations or elements in the precision matrix, but rather denote the probability of a non-zero term in the precision matrix. For brevity's sake we only show



(a) Argentina (E=1)

(b) Argentina (E=0)

(c) Colombia (E=1)

(d) Colombia (E=0)

Figure 2.5: Posterior Edge Probabilities

two countries, but the structures are similar across the remaining four. Again, these figures show

strong connections in the graphs, which support the use of our method.

2.5.3 Predictive Ability of Algorithms

In addition to the inferential treatment of model association, we are also interested in making predictions by combining the models. In contrast to the last section, where the entire dataset was used retrospectively to fit models and make inference, we consider a prospective prediction framework here. That is, we use the first six months to fit the models and learn decision rules for each set of model inputs. We make both binary predictions and probabilities corresponding to a protest occurring, which are evaluated with classification error and log-loss respectively.

Table 2.5 contains classification and log loss error rates for each of the six capital cities. From this table, we see that the predictive models perform the best in Venezuela and the worst in El Salvador. Furthermore, on the whole predicting civil unrest is difficult, but our models and the fusion framework capture most of the relevant information pertaining to unrest.

Table 2.5: Prediction Error for Capital Cities

# Country	CE	Log-Loss
Argentina	0.39	0.68
Colombia	0.43	0.68
El Salvador	0.41	0.70
Mexico	0.27	0.58
Uruguay	0.39	0.66
Venezuela	0.40	0.65

2.6 Discussion

This chapter outlines methodology for inferring association structure and making predictions (both binary and probabilistic) from a set of binary models. The challenge in this setting is that there are not existing methods available, particularly from a Bayesian perspective, to learn the association structure and make consensus predictions informed by that structure. Our method allows both inference on the model structure and predictive ability by averaging across model structures. On a basic simulation scenario our method outperforms existing methods on prediction, while also providing recovering the simulated structure of the models. The methodology is then applied to a case study predicting whether a protest will occur the next day in six capital cities across Central and South America and shows positive results at modeling unrest.

2.7 Appendix

For steps (1) - (3) parallel procedures are conducted conditioning on $Y = 1$ and $Y = 0$. For simplicity we drop the index pertaining to this conditioning.

Step 1. Latent Variables

$$\begin{aligned} \pi(R) &= \frac{\Pi_{p\in\mathcal{P}}\ \pi(R^p)}{\Pi_{s\in\mathcal{S}}\ \pi(R^s)}, \\ \text{where } \pi(R^u) &\propto |R^u|^{(\frac{|u|(|u|-1)}{2})} \left(\Pi_{i\in u}\ |R^u_{ii}|^{-\frac{|u|+1}{2}}\right). \end{aligned}$$

Denote R_{ii} as the principal submatrix of R.

$$\pi(\mu|R) \sim N(0, R)$$

Then, the joint posterior of μ and R follows as:

$$\pi(\mu, R|y) \propto \pi(R)\pi(\mu|R)\Pi_{i=1}^{n} p(y_i|\mu, R).$$

Using the multivariate probit model, we introduce latent variables Z such that:

$$\begin{aligned} \pi(\mu, R, Z|y) &\propto \pi(R)\pi(\mu|R)\pi(Z|\mu, R)\Pi_{i=1}^{n}\delta(Z_i \in B_i) \\ \text{such that } \pi(Z|y, \mu, R) &= \Pi_{i=1}^{n}\pi(Z_i|y, \mu, R) \\ \text{where } \pi(Z_i|y, \mu, R) &\propto N(\mu, R)\delta(Z_i \in B_i). \end{aligned}$$

Hence, the latent variables Z are sampled from truncated multivariate normal distributions in the Gibbs sampler.

Step. 2 G, R, and μ

Let $W = ZD$, where D is a diagonal matrix with entries $(d_1, ..., d_j)$. Let

$$\pi(D|R) = \Pi_{i=1}^{J}\, \pi(d_i|R), \text{ and } \pi(d_i^2|R) \sim IG(\frac{J+1}{2}, \frac{r^{ii}}{2}),$$

where r^{ij} is the ij element of R^{-1}. Then

$$\pi(W|\mu, R, D, G) \sim \Pi_{i=1}^{n}N(W_i; \mu D, DRD)\delta(W_i \in B_i),$$

which can be decomposed consistent with G as

$$\Pi_{i=1}^{n}\frac{\Pi_{p\in\mathcal{P}}\, N(W_i^p; \gamma^p, \Sigma^p)}{\Pi_{p\in\mathcal{P}}\, N(W_i^p; \gamma^p, \Sigma^p)},$$

where $\gamma = \mu D$, $\Sigma = DRD$, and γ^p corresponds to the elements in p. Then we wish to sample from the joint posterior distribution

$$\pi(\mu, R, D, G|y, W) = \pi(G)\pi(R|G)\pi(\mu|R, G)\pi(D|R, G)\pi(W|\mu, R, D, G),$$

which is accomplished by sampling from the joint posterior of the transformed variables Σ and γ along with G as $\pi(\Sigma, \gamma, G|W)$ and then transforming variables to the original scale. The prior for R is chosen using principles from the separation strategy detailed in Barnard et al. (2000), such that $\pi(\Sigma) \sim HIW(2, I_j)$. We jointly sample Σ and γ from the joint full conditional as

$\pi(\gamma, \Sigma|G, W) = \pi(\gamma|\Sigma, G, W)\pi(\Sigma|G, W)$, where

$$\begin{aligned}
\pi(\gamma|\Sigma, G, W) &\sim N\left(\frac{\sum_i W_i}{n+1}, \frac{\Sigma}{n+1}\right) \text{ and} \\
\pi(\Sigma|G, W) &\sim HIW(2+n, W^T W + I_j - \frac{W^T J_n W}{n+1}) \\
\text{As } \pi(\Sigma, \gamma|G, W) &\propto \Pi_i N(W_i; \gamma, \Sigma) N(\gamma; 0, \Sigma) HIW(\Sigma; 2, I_j).
\end{aligned}$$

Then denote the diagonal matrix D_* with diagonal elements $d_{ii} = \sqrt{\Sigma_{ii}^{-1}}$. It follows that R and μ can be computed as $R = D_*^{-1}\Sigma D_*^{-1}$ and $\mu = \gamma D_*^{-1}$.

The graph G is sampled using a Metropolis with Gibbs step, where two vertices are randomly selected and the edge between those vertices is proposed to be added if it does not currently exist and removed if the edge exists. Then the acceptance ratio can be computed using the distribution

$$\begin{aligned}
\pi(W|G) &= \int_{(\Sigma^{-1}|G)} \int_{\gamma} \pi(W|\Sigma, \gamma, G)\pi(\Sigma, \gamma|G) d\gamma d\Sigma \\
&= (2\pi)^{-nJ/2} \frac{h(G, 2, I_j)}{h(G, 2+n, W^T W + I_j - \frac{W^T J_n W}{n+1})},
\end{aligned}$$

where $h(G, b, K)$ are the normalizing coefficients of the hyper inverse Wishart distribution,

$$h(G, b, K) = \frac{\Pi_{p \in \mathcal{P}} |\frac{K^p}{2}|^{\frac{b+|P|+1}{2}} \Gamma_{|p|}\left(\frac{b+|P|-1}{2}\right)^{-1}}{\Pi_{S \in \mathcal{S}} |\frac{K^S}{2}|^{\frac{b+|S|+1}{2}} \Gamma_{|S|}\left(\frac{b+|S|-1}{2}\right)^{-1}}.$$

Step 3. Integrate $\int_{Z_*} p[Z|Y, \mu, R, G]dZ$

Integrating this function amounts to computing the mass in each orthant. A Monte Carlo procedure is implemented, simulating from $p[Z|Y, \mu, R, G]$ and computing the mass in each orthant.

Bibliography

Albert, J. H. and Chib, S. (1993), "Bayesian analysis of binary and polychotomous response data," *Journal of the American Statistical Association*, 88, 669–679.

Barnard, J., McCulloch, R., and Meng, X.-L. (2000), "Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage," *Statistica Sinica*, 10, 1281–1312.

Bell, R., Koren, Y., and Volinsky, C. (2010), "All together now: A perspective on the Netflix prize," *Chance*, 23, 24–24.

Breiman, L. (1996), "Bagging predictors," *Machine Learning*, 24, 123–140.

— (2001), "Random forests," *Machine Learning*, 45, 5–32.

Carvalho, C. M., Massam, H., and West, M. (2007), "Simulation of hyper-inverse Wishart distributions in graphical models," *Biometrika*, 94, 647–659.

Chib, S. and Greenberg, E. (1998), "Analysis of multivariate probit models," *Biometrika*, 85, pp. 347–361.

Gneiting, T. and Raftery, A. E. (2007), "Strictly proper scoring rules, prediction, and estimation," *Journal of the American Statistical Association*, 102, 359–378.

Hoegh, A., Leman, S., Saraf, P., and Ramakrishnan, N. (2015), "Bayesian Model Fusion for Forecasting Civil Unrest," *Technometrics*, 57, 332–340.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999), "Bayesian model averaging: a tutorial," *Statistical Science*, 382–401.

Kittler, J., Hatef, M., Duin, R. P., and Matas, J. (1998), "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20, 226–239.

Lauritzen, S. L. (1996), *Graphical Models*, Oxford: Oxford University Press.

Lewis, D. D. (1998), "Naive Bayes at forty: The independence assumption in information retrieval," in *Machine Learning: ECML-98*, Springer, pp. 4–15.

Lopez, M. J. and Matthews, G. J. (2015), "Building an NCAA men's basketball predictive model and quantifying its success," *Journal of Quantitative Analysis in Sports*, 11, 5–12.

Raftery, A. E., Gneiting, T., Balabdaoui, F., and Polakowski, M. (2005), "Using Bayesian model averaging to calibrate forecast ensembles," *Monthly Weather Review*, 133, 1155–1174.

Ramakrishnan, N., Butler, P., Muthiah, S., Self, N., Khandpur, R., Saraf, P., Wang, W., Cadena, J., Vullikanti, A., Korkmaz, G., Kuhlman, C., Marathe, A., Zhao, L., Hua, T., Chen, F., Lu, C. T., Huang, B., Srinivasan, A., Trinh, K., Getoor, L., Katz, G., Doyle, A., Ackermann, C., Zavorin, I., Ford, J., Summers, K., Fayed, Y., Arredondo, J., Gupta, D., and Mares, D. (2014), "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*, ACM, pp. 1799–1808.

Robert, C. (1995), "Simulation of truncated normal variables," *Statistics and Computing*, 5, pp. 121–125.

Robert, C. P. and Casella, G. (2005), *Monte Carlo Statistical Methods (Springer Texts in Statistics)*, Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Schapire, R. E. (1990), "The strength of weak learnability," *Machine Learning*, 5, 197–227.

Schefzik, R., Thorarinsdottir, T. L., and Gneiting, T. (2013), "Uncertainty quantification in complex simulation models using ensemble copula coupling," *Statistical Science*, 28, 616–640.

Scutari, M. (2010), "Learning Bayesian networks with the bnlearn R package," *Journal of Statistical Software*, 35.

Sloughter, J. M., Gneiting, T., and Raftery, A. E. (2010), "Probabilistic wind speed forecasting using ensembles and Bayesian model averaging," *Journal of the American Statistical Association*, 105, 25–35.

Talhouk, A., Doucet, A., and Murphy, K. (2012), "Efficient Bayesian inference for multivariate probit models with sparse inverse correlation matrices," *Journal of Computational and Graphical Statistics*, 21, 739–757.

Tibshirani, R. (1996), "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 267–288.

Yuan, M. and Lin, Y. (2006), "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68, 49–67.

Chapter 3

Bayesian Model Fusion for Forecasting Civil Unrest

ANDREW HOEGH[1,3], SCOTLAND LEMAN[1,3], PARANG SARAF[2,3], NAREN RAMAKHRISNAN[2,3]
[1]DEPARTMENT OF STATISTICS, VIRGINIA TECH, BLACKSBURG, VA 24061
[2]DEPARTMENT OF COMPUTER SCIENCE, VIRGINIA TECH, BLACKSBURG, VA 24061
[3]DISCOVERY ANALYTICS CENTER, VIRGINIA TECH, BLACKSBURG, VA 24061
PAPER PUBLISHED IN *Technometrics*

Abstract

With the rapid rise in social media, alternative news sources, and blogs, ordinary citizens have become information producers as much as information consumers. Highly charged prose, images, and videos spread virally, and stoke the embers of social unrest by alerting fellow citizens to relevant happenings and spurring them into action. We are interested in using Big Data approaches to generate forecasts of civil unrest from open source indicators. The heterogeneous nature of data coupled with the rich and diverse origins of civil unrest call for a multi-model approach to such forecasting. We present a modular approach wherein a collection of models use overlapping sources of data to independently forecast protests. Fusion of alerts into one single alert stream

becomes a key system informatics problem and we present a statistical framework to accomplish such fusion. Given an alert from one of the numerous models, the decision space for fusion has two possibilities: i) release the alert or ii) suppress the alert. Using a Bayesian decision theoretic framework, we present a fusion approach for releasing or suppressing alerts. The resulting system enables real-time decisions and more importantly tuning of precision and recall.

Keywords: Ensemble Methods, Correlated Models, Integrating Heterogenous Data, Event Modeling, Big Data

3.1 Introduction

Social unrest (protests, strikes, and occupy events) is endemic in many societies, e.g., recent news happenings in the Middle East, Southeast Asia, Latin America. It is of great interest to social scientists, policy makers, governments (both local and foreign) to forecast social unrest, including the who, why, what, and when of the protest. However, modeling human behavior can be notoriously difficult (Garson 2009; Carley 2006), and civil unrest is particularly challenging (Anderson Jr. 2006; Thron et al. 2012; Stark et al. 2010). The factors that give rise to civil unrest are multifaceted and vary depending on the type of protest. Some, such as inflation, increased taxes, and drought can be easily monitored but others such as dissatisfaction with government are harder to quantify. It is also known that many of these are but necessary conditions for a protest and typically we need a tipping point to ignite the passion necessary for mass unrest. Braha (2012) provides a parsimonious description of this phenomena, "widespread unrest arises from internal processes of positive feedback and cascading effects in the form of contagion and social diffusion over spatially interdependent regions connected through social and mass communication networks." It is thus an interesting research problem to recognize such conditions from modern media and use them for forecasting civil unrest.

In recent years, new forms of social media and communication (e.g., Twitter, blogs) have ushered in new ways for citizens to express themselves, even in countries with authoritarian governments. Sometimes these new media are simply a reflection of happenings in more traditional media (e.g., news). Other times these new media afford participants entirely new ways to organize themselves, and thus provide a conduit for expression of sentiments as well as recruitment of volunteers. For instance, during the recent June 2013 Confederations Cup in Brazil, a series of protests broke out, stemming from the lack of government spending on public infrastructure in light of the costs associated with hosting the games. Pertinent information traveled quickly through social media channels (e.g., bus fare increases) and led to both spontaneous and organized protests. Thus social media provides a rich source of information for forecasting that captures the transmission of social unrest (Hua et al. 2013).

We approach forecasting civil unrest as a *system informatics* (SI) problem wherein numerous 'big data' sources are continuously monitored 24x7 using multiple models, to generate real-time forecasts (alerts) of civil unrest happenings. Our focus is specifically on the countries of Argentina, Brazil, and Mexico, all hotbeds of civil unrest in the recent past. Due to the multifaceted nature of civil unrest in these places, it is infeasible to develop one universal model that captures all types of protest events involving many forms of leading indicators. We have demonstrated that a multi-model approach (Ramakrishnan et al. 2014, to appear) enables us to leverage the selective superiorities of different approaches to forecasting civil unrest. This turns the spotlight on the fusion problem, i.e., how to fuse the alert streams arising from multiple models into one single alert stream, ensuring that no extra alerts are issued and at the same time suppressing poor quality alerts.

Three important measures for evaluating the performance of predicting civil unrest events are precision, recall, and quality score. Precision is the proportion of issued alerts that correspond to actual events, while recall is the proportion of events that are predicted. Quality score is a measure of similarity between alerts and events. An implicit fourth measure is lead time, i.e., the amount of advance time by which a forecast 'beats the news.' We will not explicitly focus on modeling/optimizing lead time here, but our framework requires that alerts precede the corresponding news events.

A visual depiction of the entire SI framework can be seen in Figure 3.1. Each model processes source data independently to release alerts corresponding to future protests. The source data varies across models and while there may be some overlapping data sets (e.g., Twitter is used by four models), data is aggregated and used in a different manner. A parameter η is invoked to control whether fused alerts favor precision or recall. Releasing more alerts from the various models will cause drops in precision while recall will improve; thus, the statistical research question in fusion is focused on balancing this tradeoff, specifically: Should a given alert be issued or suppressed?

Given a set of alerts, the first step in the fusion framework is to cluster similar alerts from different models. Then, a classifier is developed using a latent multivariate Gaussian representation (Chib and Greenberg 1998; Albert and Chib 1993) of the model alerts in each cluster. The benefit of the

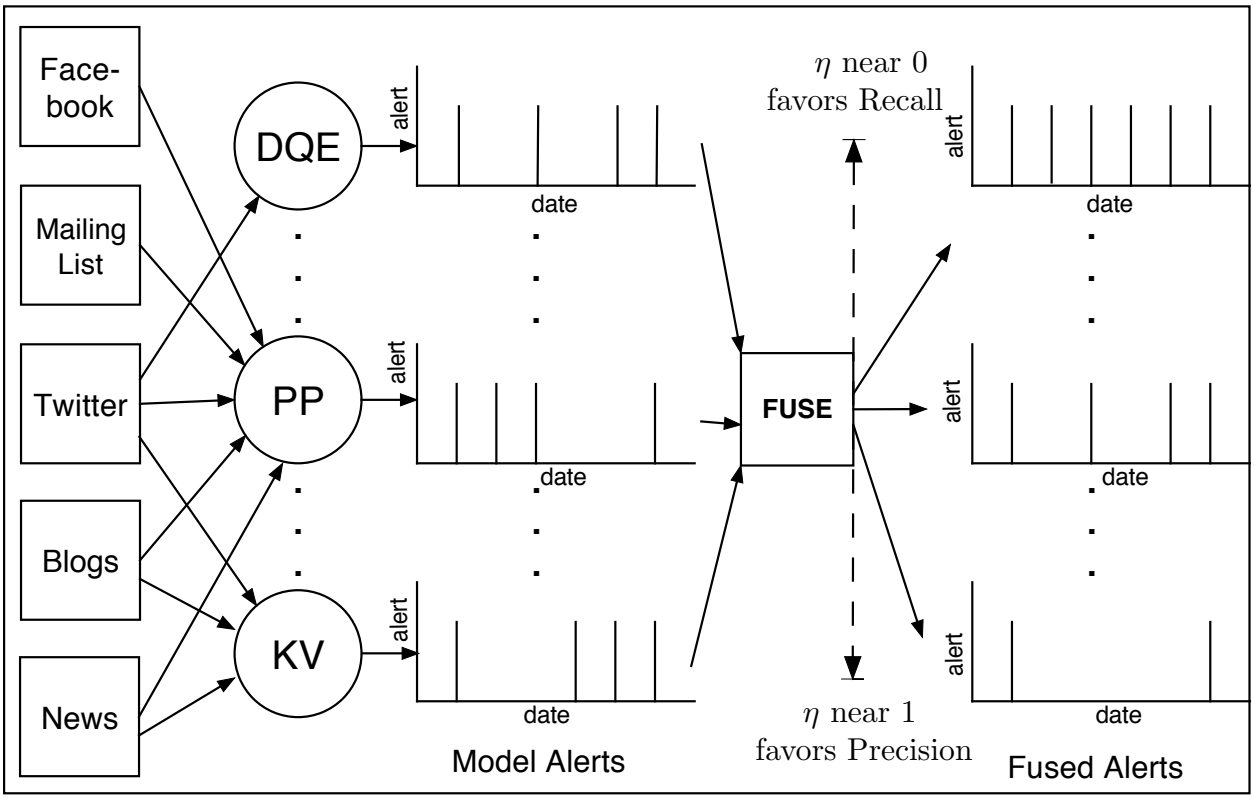


Figure 3.1: *System Informatics* framework for Bayesian model fusion, where DQE, PP, and KV are underlying models detailed in Sec 2.1 .

latent Gaussian specification is that the association structure between the models can be modeled and accounted for. Finally through a decision theoretic framework, a loss function that controls the precision-recall tradeoff determines whether to issue or suppress alerts.

The complexity of the system and the uniqueness of the problem is such that novel extensions and combinations of existing methods are required. In particular, the proposed methods contain similarities to meta-analysis, boosting, and Quadratic Discriminant Analysis (QDA); however, these methods are not directly applicable to the research problem at hand.

Statistical meta-analysis (Hedges and Olkin 1985) encompasses a broad class of techniques for combining multiple data sources. Originally such methods were developed for integrating results from multiple studies in order to create more statistically powerful inferences, or contrast

discordance in differing results. Under similar experimental conditions, Bayesian meta-analyses (Gelman and Hill 2006) are in spirit straightforward methods for fusing multiple data sources.

Another related class of techniques that are often applied to predictive fusion are boosting methods (Schapire 1990), wherein an ensemble of weak classification models form a strong unified learner/predictor. In typical cases where boosting can be applied, several models from a common model class are fit, yielding a set of predictors. By weighting and averaging each of these predictors, a single more accurate and reliable predictor is found. Such methods have been employed with great success throughout the disciplines of machine and statistical learning (Schapire and Freund 2006).

Because each of the constituting models, which we are attempting to fuse, are not conceptually similar, they do not share a common parameter space, straight-forward boosting algorithms and/or meta-analysis techniques cannot be applied. In subsequent sections of the paper, we develop a predictive fusion framework, which is completely agnostic to the models that are used for prediction. Specifically, these models need not share any parameters, or come from any common model class. Only model predictions are required.

The classification component in our fusion framework also has the flavor of QDA (Hastie et al. 2009), which we have extended for binary variables. Rather than creating a classifier on the continuous space like QDA, we integrate out the latent variables to determine optimal rules for a binary set of model inputs; this will be further illustrated in Section 3.4.4.

Our contributions include an explicit way to handle problems with massive and often unstructured data (Feldman and Sanger 2007). These rich, complicated data streams can be farmed out and analyzed separately providing a means for incorporating segmented domain knowledge or expertise. Collectively, these models extract the sufficient information to capture the dynamics driving the system. Then using our Bayesian model fusion framework, these separate models are combined in principled manner allowing enhanced control of the system. We are able to increase the precision and quality of our forecasts of civil unrest. This article is focused on predicting civil unrest, however, the general framework can be applied to a variety of problems. The fusion process is ag-

nostic to the underlying models, learning dependencies and redundancies of such models to issue predictions.

The remainder of this article is organized into five sections. Section 3.2 compares fusion with a single super-model and details the underlying models that are inputs to the fusion mechanism. Section 3.3 defines civil unrest for the purposes of this analysis and details the structure of the data including the alert-event matching scheme. Section 3.4 describes the Bayesian model fusion algorithm implemented to model civil unrest. Section 3.5 provides an overview of the case study for predicting civil unrest in Argentina, Brazil, and Mexico. Section 3.6 concludes with a discussion.

3.2 Modular Fusion vs. Super-Models

There are two schools of thought in how to design an approach to generate alerts from massive data sources. By a 'super-model' approach, we refer to an integrated algorithm for forecasting unrest that considers all possible input data sources simultaneously. One benefit of a super-model is that the joint structure between data sources can be captured, at least theoretically. For instance, Facebook and Twitter could offer mutually reinforcing signals for a given country where they both have penetration among the relevant population but can offer complementary evidence in another country where they cater to different segments of the populace. Such dependencies can be captured in a super-model approach. The obvious drawback of a super-model is the significant computational cost involved in bundling all relevant data for processing.

In contrast, the modular fusion approach is targeted at distributed processing of heterogeneous data sources, either singly or in distinct combinations. Such an approach allows segmented expertise to be naturally captured, e.g., one model could use Facebook to detect organized protests, and another could use Twitter to hunt for signals about spontaneous events (and a third model could use both). The modular fusion approach also permits easy addition (and removal) of models as new data sources are considered. Our Bayesian model fusion strategy for SI adopts this approach.

3.2.1 Models considered in this paper

Adopting the modular fusion approach, we next detail how information is extracted from unstructured social media data sources to produce discrete predictions of upcoming protests. The underlying models use open source information primarily from social media as inputs as well as a GSR (gold standard report) of historical protests organized by a third party, to produce alerts corresponding to upcoming civil unrest events. The alert structure consists of four elements: i) date of protest (day, month, year), ii) location of the protest (country, state, city), iii) event type (i.e., the reason for protest, and whether it was violent/non-violent), and iv) population (that will protest). Hence, for any particular alert, the structure is $\mathcal{A} = \{date, location, event\ type, population\}$. Thus on day t the models generate a set of alerts $\mathcal{A}_t = \{\mathcal{A}_{t_{11}}, ..., \mathcal{A}_{t_{1n_1}}, \mathcal{A}_{t_{21}} ..., \mathcal{A}_{t_{5n_5}}\}$, where $\mathcal{A}_{t_{11}}$ is the first alert issued by model 1 on day t and $\mathcal{A}_{t_{1n_1}}$ is the n_1^{th} alert issued by model 1 on day t. The fusion process takes $\mathcal{A}_t$ as an input and determines which alerts to issue and suppress.

Our modular fusion approach is intended to be agnostic to the details of the specific models, only requiring the alerts from the models and providing seamless integration of additional models. In this article we consider five models: i) historical protests, ii) planned protest, iii) spatial scan, iv) keyword volume, and v) dynamic query expansion. Next we detail how these five models extract information from datasets to produce alerts.

The historical protest (HP) model uses newspaper information about past reported protests to identify commonly recurring protests and forecasts that such protests will happen in the future as well. From the GSR, the HP model identifies frequently occuring four-tuples: (day, location, event type, population) using the last three months of protest data. Location, event type, and population are all categorical variables whose values are determined by consulting the GSR. The day is modeled as the day of the week and in issuing forecasts a fixed lead time (of 2 weeks) is used in choosing the forecast protest date. Frequently recurring event types above a specified threshold (e.g., twice per month) are issued. For instance, 'farmers protesting government taxes in Rio de Janiero, Brazil on an upcoming Wednesday', is the type of alert issued by the HP model.

The planned protest (PP) model extracts information from news, blogs, Twitter, mailing lists, and Facebook in an attempt to capture organized protests, i.e., by organizations that use social media as a way to recruit volunteers and galvanize support. This model works by extracting key phrases from text signaling intent to protest, detecting dates and locations when such protests are intended to happen. Phrases are drawn from English, Spanish, and Portugese. Examples of phrases are *preparación huelga* and *llamó a acudir a dicha movilización*. If an article or posting containing such a phrase is detected, the algorithm aims to identify a mention of a future date and a location, both proximal to the detected phrase. If a future date cannot be found (e.g., only past dates are identified) then the article is discarded. If a date mention is relative (e.g., the use of the phrase 'next Saturday') the TIMEN (Llorens et al. 1995) enrichment engine is used to convert the mention into an absolute date. As an example, an alert generated by the PP model could be the result of scraping information from a Facebook event page on an upcoming labor strike.

Like the PP model, the keyword volume (KV) model also uses a variety of news sources but tracks the daily aggregated volume of protest-related keywords, and uses such volumes in a logistic - LASSO regression model to forecast protests. Additional economic variables and Internet access statistics are also used in the model. As outlined in Ramakrishnan et al. (2014, to appear) using subject matter experts in political science and Latin American studies, we organized a dictionary of nearly 800 words and phrases that are useful indicators to track in social media. Examples of words and phrases are 'protest' and 'right to work'. The KV model uses features, i.e., volumes of these keywords, from one day to forecast if a protest will occur on the next day. Once an affirmative prediction is made, the tweets are analyzed in greater detail to determine the location, event type, and population. Location is determined using geocoding algorithms; event type and population are predicted using a naive Bayes classifier. For more details, see Ramakrishnan et al. (2014, to appear).

The spatial scan (SS) algorithm uses Twitter to track clusters of geotagged tweets enriched with protest-related words, and organizes chains of such clusters in space and time. Clusters that grow over overlapping time windows are used as indicators of upcoming protests. A fast subset scan algorithm (Neill 2012) is applied over a grid of geolocated cells to identify a relevant cluster that

is enriched with the usage of words from our protest dictionary. The cluster is then tracked over multiple time slices to determine if it persists in size (or grows or decays). A growing cluster over three consecutive slices is used as an affirmative determination of a protest. The location, event type, and population are predicted by analyzing and/or classifying the tweets constituting the cluster. The date is forecast using a simple linear regression model trained against the GSR.

The dynamic query expansion (DQE) model uses Tweets as well but is intended to be an unsupervised learning system that can detect new forms of protests hitherto not encountered (e.g., recently there were protests in Venezuela reg. toilet paper shortage and keywords used in KV and PP do not involve any relevant phrases). This model begins with a seed set of keywords and expands it to learn a dynamic set of terms that reflect ongoing chatter on social media. The expanded set is then used to construct a network of tweet nodes from which anomalous subgraphs are mined to forecast specific events. For additional details on the DQE model, readers are referred to Ramakrishnan et al. (2014, to appear).

Note that some models use just one source of data (e.g., DQE) but others (e.g., PP) use multiple sources of data. Further, note that the sources of data overlap across the models. While there may be overlap in the data sources, the models individually have selective superiorities that collectively provide a means for capturing a variety of civil unrest situations. As the models described here are a necessary component in predicting civil unrest, details and references are given so the reader can understand the underlying mechanism and implement similar models. The remainder of the article focuses primarily on the fusion process, which was designed to be modular, giving ample opportunity for segmented groups to predict civil unrest using their expertise. Hence, it is necessary for the fusion component to be flexible in integrating models by identifying dependencies and redundancies across models to issue a streamlined set of alerts.

3.3 Preliminaries

3.3.1 Defining Protests

An obvious question is: what constitutes an actual protest? Many events with a large number of people gathering to support a common case (e.g., a farmer's market or a victory parade) are not civil unrest events. The origin of the event definition lies with the Integrated Data for Events Analysis (IDEA) framework outlined in Bond et al. (2003), but is modified for our purposes. Rather than exactly identifying what constitutes a protest, it is easier to define exclusion criteria. Entertainment performances and sporting events, natural disasters, terrorists or criminal activities, general instability, and scandals are not considered civil unrest events. However, strikes by soccer players or mass protests following a natural disaster would be valid instances of protest. Another component of identifying civil unrest revolves around whether the event reaches a relevance threshold. Rather than identifying specific criteria, the determination is made as to whether the event warrants a mention in major newspapers in the country of interest. The event need not be the feature of the article, but can be discussed in the context of another story (e.g. a story about traffic mentions a protest inhibiting traffic flows). For this application, the occurrence of civil unrest events are identified by subject matter experts (disjoint from the authors) reading newspapers of record in each country and recording protests.

Once an instance of civil unrest has been identified, the event needs to categorized. For each case, five components are recorded: i) date of the protest (day, month, year), ii) date of reporting of the protest (day, month, year), iii) location of the protest (country, state, city), iv) event type (reason for protest, whether it was violent/non-violent), and v) population (that protested). The protest type includes the following categories: employment and wages, housing, energy and resources, other economic policies, other government policies, and other. The population protesting can be one of the following groups: business, ethnic, legal, education, religious, medical, media, labor, refugees/displaced, agricultural, or general population (if they do not fall into any of the other groups). The set of protests recorded by subject matter experts is referred to as the GSR. The GSR

format matches that of alerts issued by our models, with the exception that the GSR contains two dates, one pertaining to the protest date and another to when the protest is reported. Similar to the alert structure, the events are characterized as $\mathcal{E}_t = \{\mathcal{E}_{t_1}, ..., \mathcal{E}_{t_{n_t}}\}$, where the first event on day t is $\mathcal{E}_{t_1} = \{date(protest), date(reported), location, event\ type, population\}$.

The distinction between date of the protest and date of reporting is crucial. Ideally we need a forecast before the date of reporting and with a forecasted date as close to the date of the protest. Furthermore, alerts are generated daily and decisions regarding issuance or suppression also need to be made daily. However, the GSR, and thus the record of events, is only updated monthly. Therefore, an evaluation of the alerts for the previous month is made once that month's GSR is finalized. The next section describes this process, in which the date of the protest, location, event type, and population, together help define the quality in accurately forecasting events.

3.3.2 Alert-Event Matching

Because models (even a single one) issue multiple alerts and because there are multiple protest events, a necessary component of evaluating performance is a strategy for matching alert-event pairs. The first required component is a similarity measure between individual alerts and events which is known as the Quality Score (QS) and defined by a piecewise formula comparing every element of the event against the corresponding element of the alert. QS is defined to be zero under certain obvious conditions: (i) if the alert country and event country do not match, (ii) if the lead time is < 0 (i.e., the alert is issued after the event is reported), and (iii) if the forecasted date of event and actual date of event are more than seven days apart. If these exclusion conditions are not triggered, QS is defined as:

$$\begin{aligned}
QS(\mathcal{A}, \mathcal{E}) &= \text{date.score} + \text{location.score} + \text{type.score} + \text{population.score}, \quad \text{where} \\
\text{date.score} &= 1 - \frac{|date_{\mathcal{A}} - date_{\mathcal{E}}|}{7}, \\
\text{location.score} &= \frac{1}{3}\delta(country_{\mathcal{A}} = country_{\mathcal{E}}) + \frac{1}{3}\delta(state_{\mathcal{A}} = state_{\mathcal{E}}) + \frac{1}{3}\delta(city_{\mathcal{A}} = city_{\mathcal{E}}), \\
\text{type.score} &= \frac{2}{3}\delta(reason_{\mathcal{A}} = reason_{\mathcal{E}}) + \frac{1}{3}\delta(violent_{\mathcal{A}} = violent_{\mathcal{E}}) \\
\text{population.score} &= \delta(population_{\mathcal{A}} = population_{\mathcal{E}}).
\end{aligned}$$

The $\delta()$ is an indicator function. Examples of issued alerts and the corresponding matched events with the resultant QS are shown in Figure 3.2. For instance, in the second alert-event pair the

---

Alert 1: {(06/17/2013),(Argentina, Corrientes, Corrientes),(Other - not violent), (General Pop)}

Event 1: {(06/18/2013),(Argentina, Rio Negro, Villa Regina),(Other Econ. - not violent), (Agriculture)}

QS= date.score (0.86)+location.score (0.33) + type.score (0.33) + population.score (0) = 1.52

Alert 2: {(06/27/2013),(Mexico,Veracruz,Isla),(Other Gov Policy- not violent), (General Pop)}

Event 2:{(6/27/2013),(Mexico,Veracruz,Córdoba),(Other Gov Policy - not violent), (General Pop)}

QS= date.score (1)+location.score (0.67) + type.score (1) + population.score (1) = 3.67

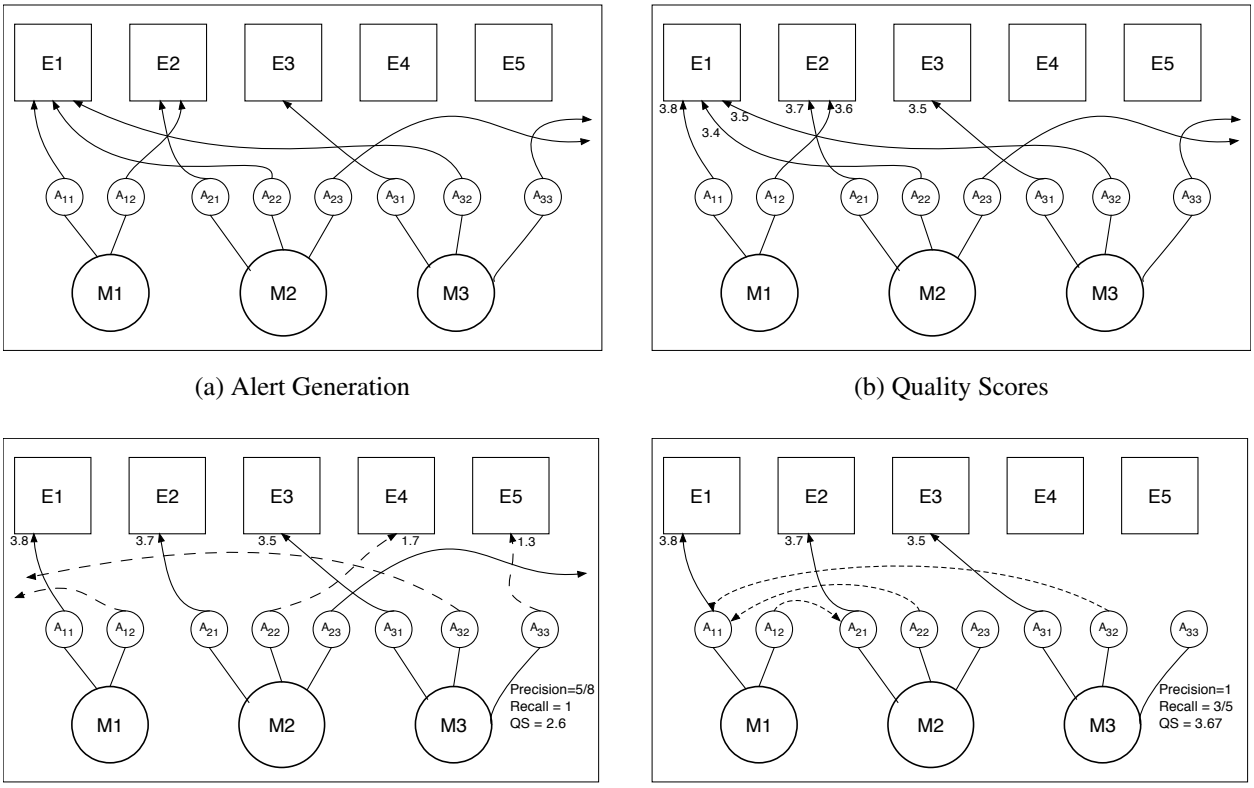
Figure 3.2: Matched alert-event pairs

resultant QS is 3.67 as the only difference between alert and event is the city. Note that QS is scaled to be in $[0, 4]$.

We can think of the QS formula as helping define the space of possible matchings between alerts and events. We annotate every matchable edge (QS > 0) between alerts and events with their QS, and compute a maximum bipartite matching to evaluate the overall performance of the set of alerts issued. We enforce the added constraint that an alert can be matched to at most one event (and vice versa). The overall quality score is then assessed based on the maximum quality score achievable in a bipartite matching between alerts and events as defined in Munkres (1957).

Alert-Event Matching Illustration

Consider the visual depiction of the process as shown in Figure 3.3. As seen in Panel (a), each model issues multiple alerts, some of which can correspond to realized events. In particular, note that an event can potentially be forecast by multiple models, but perfect identification of the event will be rare. So as Panel (b) shows, the QS that each alert achieves can be different. Additionally, alerts are also generated for events that do not occur (e.g., A_{33}). At this point two distinct scenarios are depicted in Panels (c) and (d), a case where all alerts from the models are released and another



(a) Alert Generation

(b) Quality Scores

(c) Matching without Fusion

(d) Matching with Fusion

Figure 3.3: Overview of alert-event matching.

with alerts suppressed by fusion. Whereas Panel (a) depicted potential for models to forecast events, Panel (c) presents a concrete assignment of alerts to events when all alerts are released. Dashed lines in Panel (c) are used to denote either alerts that did not match to events, or that matched fortuitously to events other than the target. This panel illustrates the two possible negative effects of issuing extraneous alerts: either alerts go unmatched, lowering precision or fortuitously match to 'incorrect' events, diminishing the mean QS. Such rogue alerts can be a result of near duplicate alerts or alerts not corresponding to an event, but the net effect is the same. Finally in Panel (d) consider a case where fusion clusters similar alerts (shown as dotted lines), while also suppressing poor alerts (no arrows). This results in lower recall, but higher precision, and higher quality scores.

In general without fusion, issuing more alerts (accomplished by incorporating more models) is not

uniformly beneficial, even if the alerts accurately predict civil unrest. In fact, too many alerts is detrimental. There are two negative effects of issuing excess alerts: alerts are unmatched causing precision to diminish or alerts are matched with an event with low quality score reducing the average quality score. Next we detail the implementation of our fusion process.

3.4 Bayesian Model Fusion

On a given day, a set of alerts, $\mathcal{A}_t$, are generated from the collection of underlying models. Each individual alert contains information pertaining to location, date of protest, population protesting, and the type of protest. Given $\mathcal{A}_t$, our fusion model consists of two stages. First, a scheme for clustering similar alerts from different models is invoked that reduces $\mathcal{A}_t$ into a set of clusters $\mathcal{M}_t$. Each cluster contains up to five alerts with no more than a single alert from each model defined as $\boldsymbol{m} = (m_1, m_2, ..., m_5)$, where $m_i \in \{0, 1\}$ denotes whether an alert from model i is included in the cluster. This results in $(2^5 - 1) = 31$ unique cluster types. Second, for each cluster type we estimate the probability of an event occurring using data from the previous month. Based on a loss function, an alert from each instance of cluster type is emitted if the probability of a match is sufficiently high, otherwise alerts in the cluster type are suppressed. The following subsections detail our clustering procedure, decision rule, and implementation of Bayesian Model Fusion.

3.4.1 Alert-Alert Clustering

The first step is to reduce extraneous alerts by clustering similar alerts. Specifically, clusters of similar alerts are formed so that a given cluster contains either zero or one alert from each model. This is done by combining alerts such that $QS(\mathcal{A}_{tij}, \mathcal{A}_{t'i'j'}) > 3.0$, where $QS()$ is the previously defined similarity score now applied to two alerts and $\mathcal{A}_{tij}$ denotes the j^{th} alert issued from the i^{th} model on day t. A QS of three is used in this analysis, but the actual QS can be viewed as a tuning parameter. The clustering effectively removes redundant alerts as the complete set of alerts $\mathcal{A}_t$ is mapped to a set of clusters $\mathcal{M}_t$,. Now rather than considering the complete set of alerts a smaller

subset is evaluated as only a single alert in each cluster can be released. The meta information of each alert is retained and given the similarity of the alerts, a single alert from each cluster is randomly selected to be considered for release.

3.4.2 Fusion Concepts

In this problem, multiple users are viewing alerts with varying utilities corresponding to individuals. To create a flexible procedure to account for individual utilities, a loss function is introduced that associates a cost with each type of decision. Given that this analysis is confined to predicting the presence or absence of events, loss functions can be expressed via Table 3.1, where $L(E, F(\boldsymbol{m}))$ denotes the loss for decision $F(\boldsymbol{m})$ and event outcome E, where $E = 1$ denotes that an event occurred.

Table 3.1: Loss Function Matrix: $L(E, F(\boldsymbol{m}))$

		Predicted	
		$F(\boldsymbol{m}) = 1$	$F(\boldsymbol{m}) = 0$
Actual	$E = 1$	c_{11}	c_{10}
	$E = 0$	c_{01}	c_{00}

The framework for tuning precision and recall is established by setting $c_{11} = c_{00} = 0$ and allowing c_{10} and c_{01} to vary. Then the parameter η is defined as:

$$\eta = \frac{c_{01}}{c_{10} + c_{01}}. \tag{3.1}$$

Values of η near one heavily penalize alerts that do not correspond to an event (or false alarms). The result is only the predictions with the highest probability of matching an event are issued; thus, precision is emphasized. Similarly with η near zero, failing to predict events would be more costly than issuing alerts not corresponding to an event. Hence, small values of η favor recall.

For a given cluster type, there are two possible decisions: issue an alert, $F(\boldsymbol{m}) = 1$, or suppress the alert, $F(\boldsymbol{m}) = 0$. The following is a standard result in Bayesian decision theory (Berger, 1985).

Proposition 1. *The set of decisions satisfying:* $F(\boldsymbol{m}) = 1$ *if* $P[E = 1|\boldsymbol{m}] > \eta$ *and* $F(\boldsymbol{m}) = 0$ *if* $P[E = 1|\boldsymbol{m}] \leq \eta$ *is optimal.*

Unfortunately, practical constraints make an optimal solution untenable. In particular, the real-time nature of this prediction exercise requires an algorithm capable of executing in a short manner of time. The computational bottleneck is the alert-event matching process described in Section 3.2. Computing $P[E = 1|\boldsymbol{m}]$ depends upon which cluster types are issuing alerts. For instance, if a single cluster type, $\boldsymbol{m}^* = (1, 0, 0, 0, 0)$, is suppressed because $P[E = 1|\boldsymbol{m}^*] < \eta$ then the matching algorithm needs to be re-run on the training data as $P[E = 1|\boldsymbol{m}]$ changes for $\boldsymbol{m} \neq \boldsymbol{m}^*$. This is because events that previously matched alerts from $\boldsymbol{m}^*$ in the training data are now free to match with alerts issued from other cluster types. Hence, computed probabilities change for the other cluster types. An optimal solution would require require a massive model selection set to calculate $P[E = 1|\boldsymbol{m}]$ for all $\boldsymbol{m}$ that considers the 2^{31} possibilities in which alerts from cluster types are issued or suppressed. The time alone required to run the 2^{31} sets of alert-event matching procedures (let alone any statistical machinery) is considerably longer than 24 hours. Hence we adopt a practical solution, estimating $P[E = 1|\boldsymbol{m}]$ when all clusters types issue alerts. This enables users to treat η as a tuning parameter, viewing results from past months to calibrate η to their respective utilities.

3.4.3 Data Generation Mechanism

In order to suppress lower quality alerts, it is necessary to learn $P[E = 1|\boldsymbol{m}]$. A Bayesian classifier is developed using the underlying data generation mechanism for $P[\boldsymbol{m}|E]$. That is, the process by which models issue alerts in the presence or absence of an event is modeled and flipped into a classifier via Bayes' rule. To capture model dependencies, latent multivariate Gaussian variables

$\{\boldsymbol{Z}_1, \boldsymbol{Z}_0\}$ are introduced as:

$$E = 1: \qquad \mathbf{Z_1} \sim N(\boldsymbol{\mu}_1, \mathbf{R}_1) \tag{3.2}$$

$$E = 0: \qquad \mathbf{Z_0} \sim N(\boldsymbol{\mu}_0, \mathbf{R}_0) \tag{3.3}$$

where the latent variables $\mathbf{Z}_1' = (z_{11}z_{12}, ..., z_{1i})$, $\mathbf{Z}_0' = (z_{01}, z_{02}, ..., z_{0i})$ are truncated such that $z_{ij} > 0$ if $m_j = 1$ and $z_{ij} < 0$ if $m_j = 0$. The covariance matrices are constrained to correlation matrices due to identifiability issues as in Albert and Chib (1993). Given $p[\boldsymbol{Z}|E = i, \boldsymbol{\mu}_i, \boldsymbol{R}_i]$ we can obtain, $P[\boldsymbol{m}|E = i, \boldsymbol{\mu}_i, \boldsymbol{R}_i]$ by integrating out the latent variables,

$$P[\boldsymbol{m}|E = i, \boldsymbol{\mu}_i, \boldsymbol{R}_i] \;=\; \int_{Z^*} (2\pi)^{-mn/2} |\boldsymbol{R}_i|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{Z} - \boldsymbol{\mu}_i)' \boldsymbol{R}_i^{-1}(\mathbf{Z} - \boldsymbol{\mu}_i)) d\boldsymbol{Z}, \tag{3.4}$$

where $\boldsymbol{Z}_*$ denotes the orthants such that $z_{*j} > 0$ if $m_j = 1$ and $z_{*j} < 0$ if $m_j = 0$. The integration of (3.4) can be computed using a simple accept-reject sampling method (Robert and Casella 2005). This integration is similar to QDA, but with binary data rather than continuous. QDA establishes a decision boundary between the two distributions, whereas our procedure computes a probability for each orthant corresponding to the discrete model inputs.

3.4.4 Fusion Decision

The goal is to learn $P[E = 1|\boldsymbol{m}]$, which enables tuning of precision and recall by issuing or suppressing alerts according to the loss defined by the parameter η in (3.1). This posterior distribution can be computed as

$$P[E = 1|\boldsymbol{m}] = \int \frac{P[\boldsymbol{m}|E = 1, \boldsymbol{\mu}_1, \boldsymbol{R}_1]P[E = 1]}{P[\boldsymbol{m}|E = 1, \boldsymbol{\mu}_1, \boldsymbol{R}_1]P[E = 1] + P[\boldsymbol{m}|E = 0, \boldsymbol{\mu}_0, \boldsymbol{R}_0]P[E = 0]} p(\boldsymbol{\Theta}) d\boldsymbol{\Theta}, \tag{3.5}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{R}_1, \boldsymbol{\mu}_1, \boldsymbol{R}_0, \boldsymbol{\mu}_0\}$. Priors for $\boldsymbol{\Theta}$ are drawn from Talhouk et al. (2012) in order to use the Parameter eXpansion and Data Augmentation (PXDA) approach detailed therein:

$$P(\boldsymbol{\mu}|\boldsymbol{R}) \sim N(0, \boldsymbol{R}) \text{ and } p(\boldsymbol{R}) \propto |\boldsymbol{R}|^{\frac{M(M-1)}{2}-1} (\prod_{i=1}^{M} |\boldsymbol{R}_{ii}|)^{-(M+1)/2},$$

where $\boldsymbol{R}_{ii}$ is the principal submatrix of $\boldsymbol{R}$. We use a naive Bayes type prior in which $P[E = 1] = \sum_{j=1}^{n} \delta(E_j = 1)/n$, where n is the number of points in the training data set and $\delta(E_j = 1)$ in an indicator function that the j^{th} data point was a realized event.

For general details on PXDA readers are referred to Liu and Wu (1999) and van Dyk and Meng (2001). As conjugate priors do not exist on correlation matrices, PXDA allows an augmentation converting the correlation matrix $\boldsymbol{R}$ into a valid covariance matrix. This allows Gibbs sampling of both a covariance matrix from an inverse Wishart distribution and a transformed mean from a normal distribution. Finally, these are converted back to the original mean and correlation scale. Tuning proposal densities for sampling matrices can be a cumbersome procedure, so circumventing that via a Gibbs sampler improves the efficiency of the algorithm. The MCMC procedure implemented for inference can be found in the Appendix.

3.5 Bayesian Model Fusion for Civil Unrest

3.5.1 Overview

Using data from May 2013 to train the Bayesian model fusion framework, alerts are issued and evaluated for June 2013. Predictions are made for the countries of Argentina, Brazil, and Mexico. As mentioned previously, June 2013 was a month in which numerous protests broke out in Brazil. Historical trends alone would not be sufficient to predict the sheer number of protests, so the performance during this month shows the efficacy of mining social media data to predict protests. For reference, in Brazil alone 74 events took place during May 2013. This increased in June to 425 events. Data and R code for the algorithms in this article can be found in the supplementary files.

3.5.2 Model Estimation

Convergence occurs very rapidly with a burn in period of less than 50 iterations. This implementation uses 5000 iterations for the MCMC and 100,000 Monte Carlo samples for the accept/reject sampler built into each iteration. The algorithm runs on a MacBook Pro (2GHz Intel Core i7 processor, 8 GB SDRAM) in about 300 minutes. The algorithm only need be recalibrated when a new event history is added to the GSR (monthly) or when an underlying model is changed. However, should model changes be made, the fusion implementation needs to be retrained and capable of producing updated predictions in a single day.

Marginal posterior means and credible intervals for the mean parameters of the latent representation, (3.2) and (3.3), can be seen in Table 3.2. As there are twenty correlation parameters, these are omitted for brevity's sake. Collectively the probability of matching an alert for a given cluster

Table 3.2: Posterior Means and 95 percent credible intervals

Model	μ_1	$95\%CI$	μ_0	$95\%CI$
Planned Protest	-0.05	(-0.20,0.09)	-1.82	(-1.91,-1.74)
Spatial Scan	-1.34	(-1.55,-1.13)	-2.46	(-2.61,-2.31)
Historical Protest	0.22	(0.02,0.43)	-2.23	(-2.48,-2.02)
Dynamic Query Extraction	-0.03	(-0.23,0.21)	-2.14	(-2.29,-1.97)
Keyword Volume	-0.98	(-1.24,-0.68)	-2.14	(-2.36,-1.94)

type, $P[\boldsymbol{m}|E = i, \boldsymbol{\mu}, \boldsymbol{R}]$, is computed via (3.4). The output of the MCMC procedure necessary for fusion decisions are the posterior probabilities $P[E = 1|\boldsymbol{m}]$. Table 3.3 shows the posterior mean of these distributions for select model combinations, which is used to determine which alerts to issue.

For instance, the cluster type only containing a PP alert, $\boldsymbol{m} = (1, 0, 0, 0, 0)$, and the cluster only containing a SS alert, $\boldsymbol{m} = (0, 1, 0, 0, 0)$, models both have relatively weak evidence of an event without agreement of other models, with $P[E = 1|(\boldsymbol{m} = (1, 0, 0, 0, 1))] = 0.21$ and $P[E = 1|(\boldsymbol{m} = (0, 1, 0, 1, 0))] = 0.24$. However, in cases where both models jointly issue an alert the

Table 3.3: Posterior Mean for $P[E = 1|\boldsymbol{m}]$

PP	SS	H	DQE	KV	$P[E = 1\|\boldsymbol{m}]$
1	0	0	0	0	0.21
0	1	0	0	0	0.24
0	0	1	0	0	0.75
0	0	0	1	0	0.53
0	0	0	0	1	0.06
1	1	0	0	0	0.51
0	0	0	1	1	0.59
1	1	0	1	0	0.74
1	0	1	1	0	0.90
1	0	1	0	1	0.70

evidence of an event increases to $P[E = 1|(\boldsymbol{m} = (1, 1, 0, 0, 0))] = 0.51$. Dependent on the η value specified by the loss function, a likely scenario is the alert in the PP cluster would initially be suppressed, but once the SS model issued a similar alert that was clustered with the PP alert, then the probability would be upgraded and an alert would be issued.

3.5.3 Results

Realistically, evaluating prediction requires considering precision, recall, and quality score jointly. The alert-event matching algorithm is a necessary evil of the process, in which poor, uncalibrated alerts are often "matched" albeit with low quality scores. This phenomenon is shown in Panel (c) of Figure 3.3, where a duplicate alert from model 2, A_{22}, matches event 4 and a rogue alert from model 3, A_{33}, matches event 6. Similarly, the first alert-event pair in Figure 3.2 represents a poor match with QS = 1.52. In both cases the alerts should have been suppressed.

To further illustrate the value of this tuning approach, six different η values are selected that span the range of computed probabilities. These levels, as well as the unconstrained instance in which all alerts are issued, are compared. A description of these levels can be seen in Table 3.4. Using the

Table 3.4: Precision Recall Tuning Levels

Level	Criteria
1	Issue all alerts
2	$\eta = 0$: cluster similar alerts
3	$\eta = 0.21$
4	$\eta = 0.51$
5	$\eta = 0.65$
6	$\eta = 0.74$
7	$\eta = 0.80$

seven levels, different numbers of alerts are issued for each level. Plots displaying the precision and recall for each of the seven tuning levels can be seen in Figure 3.4. With a large η, fusion

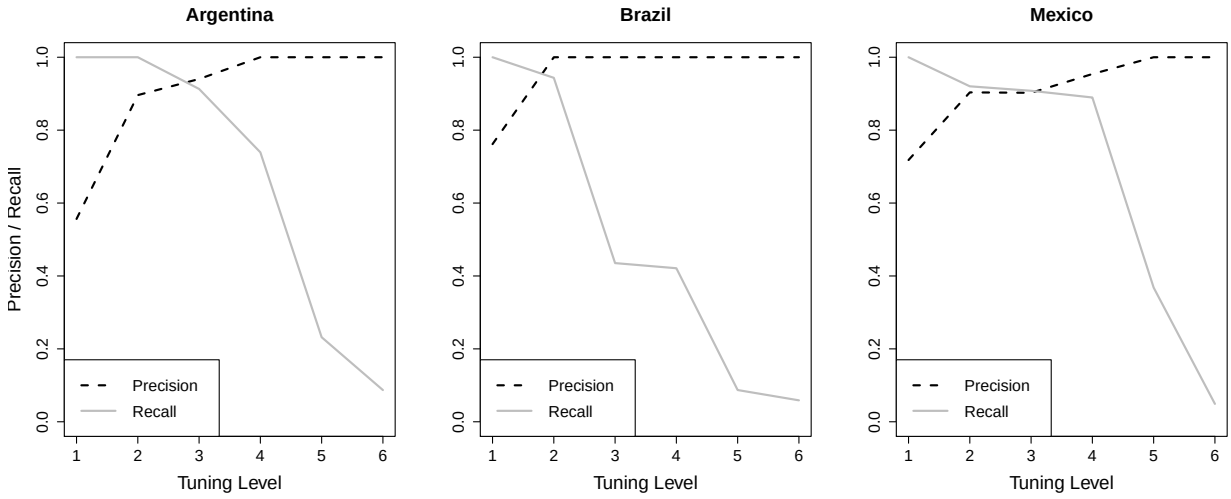


Figure 3.4: Precision Recall tradeoff by the tuning levels defined in Table 3.4

increases precision; moreover, loss functions favoring precision will also tend to have higher QS. Intuitively by suppressing alerts with a lower probability of matching an event, alerts that result in low quality matches are also suppressed. As we have shown extra alerts either match with low QS or are not matched at all. While this isn't explicitly a component of the loss function, it is an

important feature of our fusion algorithm. The mean quality score of matched alerts for each of the tuning levels can be seen in Figure 3.5. By suppressing the lowest quality alerts at each tuning

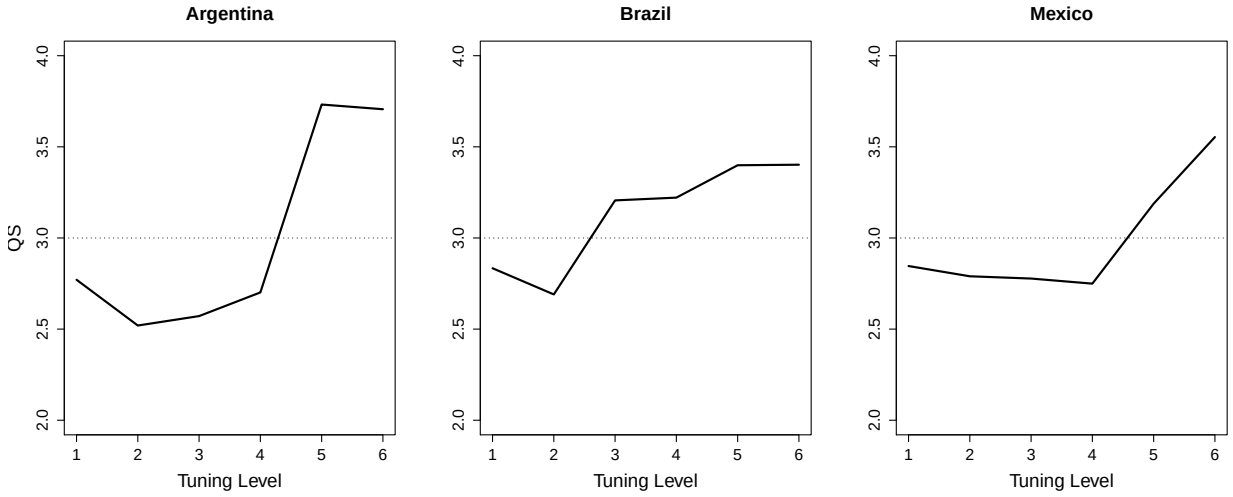


Figure 3.5: Mean QS in solid line, dashed line represents target QS of 3.0.

level, the overall mean QS increases well above the target value of 3.0 for each country.

Another way to consider quality score is to look at the distribution of the quality scores amongst the matched alert-event pairs rather then the overall mean QS. Kernel density estimates (KDE) for the mean QS combined across the three countries can be seen in Figure 3.6. In this figure, notice the distribution shifts considerably toward higher quality alerts at higher tuning levels. The fusion mechanism effectively suppresses the lower quality alerts, particularly those with a QS less than 2.

3.6 Discussion

In this article, we have presented a modular fusion approach to predicting civil unrest. The framework allows separate models to be constructed with differing selective superiorities, which is extremely beneficial when dealing with big unstructured data sets, such as the set of publicly available social media information in a country for a given month. This also allows data to be aggregated in different ways, such that the signal of civil unrest can be captured from social media. However, this

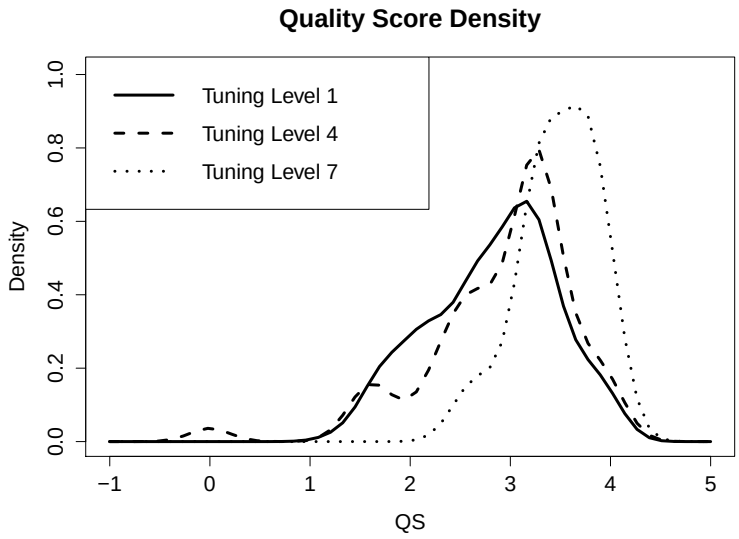


Figure 3.6: KDE of mean QS combined for all three countries for three specified tuning levels.

modular approach also presents some challenges. Creating additional models or more specifically releasing extra alerts, even strong performing ones, results in degraded precision and quality score. Our fusion framework presents a solution to combining model alerts and issuing predictions such that precision and recall tradeoffs are handled in an optimal way. While not explicitly optimizing QS, our fusion framework has been shown to retain the strongest alerts resulting in elevated overall QS.

A key feature of our approach is the tunability aspect, i.e., wherein an analyst can choose to focus on precision or recall. Recall is more appropriate (and precision less important) if an analyst uses our framework as a step in an *analytic triage*, i.e., to determine potential hotspots of civil unrest and then use domain knowledge to narrow them down to areas of interest. Precision is more appropriate (and conversely, recall is less important) if the analyst is focused on exploring a specific social science hypothesis, and thus obtaining accurate (but fewer) forecasts is desirable.

One reason that fusion is necessary is that models release similar alerts. It may be the case that the models pick up on the same signal from the same or a different data set. Irregardless of how the process arises, the net result of the similar alerts is degraded QS and lower precision. The

problem can be understood though the idea of correlated models. This is expressed through the correlation matrices in (3.2) and (3.3). An alternative to sampling the correlation matrices would be to use Gaussian graphical models to explicitly capture the association structure between the models. Graphs could be sampled during each iteration of the MCMC sampler, providing a means to visualize and estimate model dependencies. However, the association structure of the models is only tangentially related to our research question. The graph structures would be integrated out, to obtain the probabilities of identifying an event. Therefore, we sample unconstrained correlation matrices rather than the graphs.

Moving forward, a direct approach to consider the joint structure of precision, recall, and QS is being developed. Specifically, a loss function can be invoked that not only penalizes unmatched alerts and events, but also penalizes matches with QS less than some specified value. A related approach is to solve a constrained optimization for QS (e.g. maximize QS subject to precision and recall greater than some threshold). To achieve these aims, a few modifications are made to the work presented here: i) a more sophisticated model based approach for matching alerts via a Dirichlet process introducing clusters of alerts, ii) ensemble methods are used to combine information between alerts in a cluster- that is alerts that are matched are combined rather than suppressing the extra alerts, iii) given the flexibility in matching alerts, multiple alerts are allowed to be emitted from a cluster of alerts.

APPENDIX

Estimation is conducted using an MCMC procedure as follows:

1. Sample $(\boldsymbol{Z}_1|E = 1, \boldsymbol{\mu}_1, \boldsymbol{R}_1)$ and $(\boldsymbol{Z}_0|E = 0, \boldsymbol{\mu}_0, \boldsymbol{R}_0) \sim N(\boldsymbol{\mu}_i, \boldsymbol{R}_i)$. For computational efficiency the multivariate densities can be decomposed into univariate normal densities, then sampling proceeds using importance sampling as in Robert (1995). The use of standard accept-reject samplers or importance sampling with a multivariate (or univariate) normal distribution is unfeasibly slow, due to truncated regions having extremely low probabilities.

2. Sample $(\boldsymbol{\mu}_1, \boldsymbol{R}_1|\boldsymbol{Z}_1)$ and $(\boldsymbol{\mu}_0, \boldsymbol{R}_0|\boldsymbol{Z}_0)$ using PXDA as in Talhouk et al. (2012).

   (a) Sample $(d_{ii}|\boldsymbol{R}) \sim IG((M+1)/2, r^{ii}/2)$, where M is the number of models and r^{ii} are the diagonal elements of $\boldsymbol{R}^{-1}$. Then let $\boldsymbol{D}$ be a matrix with diagonal elements d_{ii} and compute $\boldsymbol{W} = \boldsymbol{Z}\boldsymbol{D}$. Then $\boldsymbol{\Sigma} = \boldsymbol{D}\boldsymbol{R}\boldsymbol{D}$ and $\boldsymbol{\gamma} = \boldsymbol{\mu}\boldsymbol{D}$.

   (b) Sample $(\boldsymbol{\Sigma}|\boldsymbol{W})$ from $IW\left(2+n, \boldsymbol{W}'\boldsymbol{W} + \boldsymbol{I} - \boldsymbol{W}' J_n \boldsymbol{W}/(n+1))\right)$, where

$$\boldsymbol{\Sigma} \sim IW(\nu, \boldsymbol{\Omega}) \propto |\frac{\boldsymbol{\Omega}}{2}|^{(\frac{\nu+M-1}{2})} |\boldsymbol{\Sigma}|^{(\frac{-\nu+2M}{2})} \exp(-\frac{1}{2} tr[\boldsymbol{\Sigma}^{-1}\boldsymbol{\Omega}]),$$

   and J_n is a $n \times n$ matrix of 1s.

   (c) Sample $(\boldsymbol{\gamma}|\boldsymbol{W}, \boldsymbol{\Sigma}) \sim N\left(\mathbf{1}_n' \boldsymbol{W}/(n+1), \Sigma/(n+1))\right)$.

   (d) Compute $\boldsymbol{R} = \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}$ and $\boldsymbol{\mu} = \boldsymbol{\gamma}\boldsymbol{Q}$, where $\boldsymbol{Q}$ is a diagonal matrix with diagonal elements $q_{ii} = \sigma_{ii}^{-1/2}$, where $\sigma_{ii}^{-1/2}$ are the diagonal elements of the precision matrix Σ^{-1}.

3. Compute $P[\boldsymbol{m}|E=1, \boldsymbol{\mu}_1, \boldsymbol{R}_1]$ and $P[\boldsymbol{m}|E=0, \boldsymbol{\mu}_0, \boldsymbol{R}_0]$ as in (3.4) for each of the cluster types $\boldsymbol{m}$.

4. $P[E=1|\boldsymbol{m}, \boldsymbol{\mu}_1, \boldsymbol{R}_1, \boldsymbol{\mu}_0, \boldsymbol{R}_0]$ is computed via (3.5), where each iteration of the MCMC sampler integrates over $\{\boldsymbol{\mu}_1, \boldsymbol{R}_1, \boldsymbol{\mu}_0, \boldsymbol{R}_0\}$ to obtain $P[E=1|\boldsymbol{m}]$ for $\boldsymbol{m} \in \mathcal{M}$.

Bibliography

Albert, J. H. and Chib, S. (1993), "Bayesian analysis of binary and polychotomous response data," *Journal of the American Statistical Association*, 88, pp. 669–679.

Anderson Jr., E. G. (2006), "A preliminary system dynamics model of insurgency management: The Anglo-Irish war of 1916-1921 as a case study," *Paper presented at the 2006 International System Dynamics Conference*.

Berger, J. O. (1985), *Statistical Decision Theory and Bayesian Analysis*, Springer.

Bond, D., Bond, J., Oh, C., Jenkins, J. C., and Taylor, C. L. (2003), "Integrated Data for Events Analysis (IDEA): An event typology for automated events data development," *Journal of Peace Research*, 40, 733–745.

Braha, D. (2012), "Global civil unrest: Contagion, self-organization, and prediction," *PLoS ONE*, 7(10), e48596. doi:10.1371/journal.pone.0048596.

Carley, K. (2006), "Destabilization of covert networks," *Computational & Mathematical Organization Theory*, 12, 51–66.

Chib, S. and Greenberg, E. (1998), "Analysis of multivariate probit models," *Biometrika*, 85, pp. 347–361.

Feldman, R. and Sanger, J. (2007), *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*, Cambridge University Press.

Garson, G. D. (2009), "Computerized simulation in the social sciences: A survey and evaluation," *Simulation & Gaming*, 40, pp. 267–279.

Gelman, A. and Hill, J. (2006), *Data Analysis Using Regression and Hierarchical/Multilevel Models*, Cambridge University Press.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009), *The Elements of Statistical Learning: Data mining, Inference, and Prediction*, New York: Springer-Verlag.

Hedges, L. V. and Olkin, I. (1985), *Statistical Methods for Meta-Analysis*, Boston: Academic Press.

Hua, T., Lu, C.-T., Ramakrishnan, N., Chen, F., Arredondo, J., Mares, D., and Summers, K. (2013), "Analyzing civil unrest through social media," *Computer*, 46, 0080–84.

Liu, J. S. and Wu, Y. N. (1999), "Parameter expansion for data augmentation," *Journal of the American Statistical Association*, 94, 1264–1274.

Llorens, H., L.Derczynski, Gaizauskas, R., and Saquete, E. (1995), "TIMEN: An open temporal expression normalisation resource." *In LREC*, 5, 3044 – 3041.

Munkres, J. (1957), "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 5, 32–38.

Neill, D. B. (2012), "Fast subset scan for spatial pattern detection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74, 337–360.

Ramakrishnan, N., Butler, P., Muthiah, S., Self, N., Khandpur, R., Wang, W., Cadena, J., Vullikanti, A., Korkmaz, G., Kuhlman, C., Marathe, A., Zhao, L., Hua, T., Chen, F., Lu, C., Huang, B., Srinivasan, A., Trinh, K., Getoor, L., Katz, G., Doyle, A., Ackermann, C., Zavorin, I., Ford, J., Summers, K., Fayed, Y., Arredondo, J., Gupta, D., and Mares, D. (2014, to appear), "Beating the news with EMBERS: Forecasting Civil Unrest using Open Source Indicators," *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Robert, C. (1995), "Simulation of truncated normal variables," *Statistics and Computing*, 5, pp. 121–125.

Robert, C. P. and Casella, G. (2005), *Monte Carlo Statistical Methods (Springer Texts in Statistics)*, Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Schapire, R. E. (1990), "The strength of weak learnability," *Machine Learning*, 5, 197–227.

Schapire, R. E. and Freund, Y. (2006), *Boosting: Foundations and Algorithms*, Adaptive Computation and Machine Learning Series, MIT Press.

Stark, O., Hyll, W., and Behrens, D. (2010), "Gauging the potential for social unrest," *Public Choice*, 143, 229–236.

Talhouk, A., Doucet, A., and Murphy, K. (2012), "Efficient Bayesian inference for multivariate probit models with sparse inverse correlation matrices," *Journal of Computational and Graphical Statistics*, 21, 739–757.

Thron, C., Salerno, J., Kwiat, A., Dexter, P., and Smith, J. (2012), "Modeling South African service protests using the national operational environment model," in *Social Computing, Behavioral-Cultural Modeling and Prediction*, Springer, pp. 298–305.

van Dyk, D. A. and Meng, X.-L. (2001), "The art of data augmentation," *Journal of Computational and Graphical Statistics*, 10, 1–50.

Chapter 4

Spatiotemporal Model Fusion: Multiscale Modeling of Civil Unrest

ANDREW HOEGH, MARCO A. R. FERREIRA, AND SCOTLAND LEMAN
DEPARTMENT OF STATISTICS, VIRGINIA TECH, BLACKSBURG, VA 24061
PAPER PUBLISHED IN JRSS: SERIES C (APPLIED STATISTICS)

Abstract

Civil unrest is a complicated, multifaceted social phenomenon that is difficult to forecast. Relevant data for predicting future protests consist of a massive set of heterogenous data sources, primarily from social media. Using a modular approach to extract pertinent information from disparate data sources, we develop a spatiotemporal multiscale framework to fuse predictions from algorithms mining social media. This novel multiscale spatiotemporal model is developed to satisfy four essential requirements: (1) be scalable to handle massive spatiotemporal datasets, (2) incorporate hierarchical predictions, (3) accommodate predictions of differing quality and uncertainty, and (4) be flexible, allowing revisions to existing algorithms and the addition of new algorithms. This article details the challenges posed by these four requirements and outlines the benefits of our novel multiscale spatiotemporal model relative to existing methods. In particular, our multiscale approach coupled with an efficient sequential Monte Carlo framework enables scalable, rapid com-

putation of richly specified Bayesian hierarchical models for spatiotemporal data.

Keywords: Sequential Monte Carlo, Spatiotemporal Modeling, Areal Data, Multiscale Modeling

4.1 Introduction

With the advent of social media and the twenty-four hour news cycle, publicized instances of civil unrest are seemingly more prevalent than ever before. Despite this, prediction of the occurrence of protests and understanding of the cascading effects of protests remains a substantial challenge. Some protests can incite more protests in a self-exciting manner in both space and time such as a Hawkes process, while other protests may be localized events. Although certain conditions need to be present for unrest, these conditions alone do not guarantee protest. Factors such as frustration with government corruption and increased taxes are certainly related to protests, but these conditions on their own are not sufficient for widespread protests. Rather, a spark is needed to galvanize citizens and cause widespread protesting. Thus, datasets that track measures such as inflation or increased bus fares are relevant but not sufficient for predicting civil unrest. Also needed is the ability to mine correspondence between citizens that incite the spread of unrest. Social media provides a conduit for these interactions and also is a rich dataset - provided the signal can adequately be filtered from all the noise.

Computation is an essential consideration for modeling and analyzing modern datasets. One approach that is growing in popularity is the use of multiscale methods, see for instance Ferreira and Lee (2007). Multiscale methods allow decomposition of data in a similar manner to wavelet basis functions (Kolaczyk, 1999). In spirit, multiscale methods are similar to treed Gaussian processes (Gramacy and Lee, 2008) in which the data are partitioned into smaller conditionally independent sections. Developments of multiscale methods to spatiotemporal data have largely focused on point referenced or Gaussian areal data (Ferreira et al., 2010, 2011). Recently, Fonseca and Ferreira (2016) created a framework for spatiotemporal Poisson data. However, this model only admits covariates at the top level of the hierarchy. Our framework contains information pertaining to multiple levels of spatial structure; thus, a framework for handling covariates at multiple levels is necessary. Hence, with this work we develop a spatiotemporal multiscale framework for areal count data that admits covariates at each level of the hierarchy.

With the increasing size and complexity of spatiotemporal datasets, the need for efficient, scal-

able algorithms grows. For instance, two popular ideas for point referenced data are covariance tapering, (Furrer et al., 2006) wherein the covariance matrix controlling the spatial structure between points is tapered for easier computation, and predictive processes (Banerjee et al., 2008), which select a subset of locations to designate as knots for computational expediency. For areal data, Conditional Autoregressive (CAR) and Intrinsic Autoregressive methods (Besag, 1974) are traditionally used for encoding spatial structure. However, with a dataset such as ours, that is intrinsically multiscale (e.g. Country/Region/State) and in which spatial correlation may propagate across each of these levels, multiple layers of spatial encoding are required in a CAR framework. Unfortunately this type of model does not scale well with larger datasets. By coupling an efficient Sequential Monte Carlo (SMC) algorithm with our novel spatiotemporal multiscale model we achieve rapid, scalable computation for spatiotemporal areal datasets.

4.1.1 Model Fusion

While 'big data' has made its way into the public vernacular, a less publicized problem is the heterogeneity of source data composing big datasets. In addition to the physical size of large datasets, the data are often compiled from different sources and are of different types making easy integration, even discounting the physical size, infeasible. Sometimes, the data can be reduced and analyzed in a single model, but other times using a collection of models may be a superior strategy (Ramakrishnan et al., 2014; Bliznyuk et al., 2014). By nearly any measure, the complete set of publicly available social media information in a given country is the epitome of a 'big' dataset. As such, collectively processing all available social media information across platforms is unfeasible, so we adopt a modular structure in which separate predictive algorithms are constructed, which utilize only a subset of the complete data. This modular approach requires a mechanism for principled integration, or fusion, of output from the separate algorithms.

Consider a process where independent models, or predictive algorithms, are constructed so that each can use different but potentially overlapping subsets of data to make predictions. It may be the case that the predictive algorithms have selective superiorities (Brodley, 1993) - in that they

are best for some tasks or areas but not all. This leaves the question of how to integrate these predictions in the model fusion step, which is the focus of this paper. In spirit, a methodology similar to that of stacking (Wolpert, 1992) is required to fuse output from the modular algorithms. Similar to the guidelines detailed in Banks et al. (2012) for syndromic surveillance systems, we develop guidelines for spatiotemporal model fusion for predicting civil unrest, which we detail below.

1. *Be scalable:* Methods need to scale with space and time to handle datasets of increasing size and complexity. Furthermore, efficient computation is desired for online estimation.

2. *Incorporate hierarchical predictions:* The modular approach yields point predictions for protests of a hierarchical nature on areal data. Hence, our methods need to integrate these predictions while respecting the spatial integrity of the data.

3. *Accommodate uncertain predictions:* While predictions are made for a specific point in space and time, uncertainty is inherent in these predictions. Hence, a smoothing mechanism is required to handle these uncertainties. Furthermore, algorithms are expected to have selective superiorities that will result in a range of prediction qualities for differing regions and event types.

4. *Be flexible:* The nature of the modular approach allows segmented expertise or groups to develop algorithms for predicting civil unrest. The fusion component needs to be agnostic to the predictive algorithm methodologies. Algorithms may be modified and new algorithms may be created. As such, we desire a flexible procedure that provides easy integration of new or improved algorithms.

Existing methods do not satisfy these four requirements. Hence, we develop a spatiotemporal multiscale model in order to meet these requirements. Our fusion framework satisfies these guidelines for efficient, scalable computation of richly parameterized spatiotemporal model fusion. While these methods are tailored to prediction of civil unrest, the principles of spatiotemporal model fu-

sion are generalizable to other applications such as disease surveillance, prediction of crime, and combination of computer models, such as those commonly used in climate modeling.

The remainder of this article is as follows: Section 4.2 outlines the data relevant for predicting unrest, Section 4.3 details multiscale partition and factorization, Section 4.4 describes our novel multiscale temporal evolution, Section 4.5 contains efficient SMC algorithms for computation, Section 4.6 compares predictions from our multiscale model with competing methods and details our results predicting civil unrest, and Section 4.7 concludes with a discussion. Additionally the appendix contains algorithms for both SMC and MCMC implementations of our model and the multiscale partitions used in this work.

4.2 Data

We seek to predict the weekly count of upcoming civil unrest events by fusing predictions from underlying algorithms designed to extract information from social media. Specifically we make one week ahead predictions for the number of protests at the state or province level across ten countries in Central and South America. In addition to historical protest counts, discrete predictions, or alerts, issued by a collection of algorithms are used. Alerts from these algorithms are designed to predict civil unrest using varying sources of social media information and methodologies. These predictive algorithms are constructed by groups different from the authors of this article; and hence, we construct our models by conditioning on these predictions. Inclusion of more accurate algorithms would improve the overarching predictability of the framework; however, our goal is a flexible, computationally feasible methodology for fusing the predictions that is flexible to adapt to changes to the set of predictive algorithms. This allows the fusion framework to be agnostic to the construction of the algorithms only considering the output. This approach provides easy modularization of big data prediction problems, in which algorithms can be developed in parallel (perhaps by different researchers or tailored with different selective superiorities) and then fused together in a principled manner.

4.2.1 Civil Unrest Data

Civil unrest occurrences are tracked over ten countries in Central and South America: Argentina, Brazil, Chile, Colombia, El Salvador, Ecuador, Mexico, Paraguay, Uruguay, and Venezuela. Protest events are identified by searching for mention of civil unrest in the two largest newspapers in each country. For an identified event, five pieces of information are obtained: the date the protest occurred, the date the protest is reported, the location as a country/state/city triplet, the event type, and the population protesting. For purposes of this work we are only concerned with the country and state level as well as the date of the protest, but ongoing work focuses on the information regarding event type and population as a multivariate extension of the models detailed herein. Thus for the remainder of the article we will focus exclusively on the spatiotemporal components of the civil unrest events.

Figure 4.1 displays weekly event counts aggregated at the national level for four countries from January 2013 through June 2014. From this figure, notice that traditional time series methods alone will be unable to account for the abrupt spikes in protests. For instance the large spike in protests in Weeks 25 and 26 in Brazil corresponds to a wave of protests that developed around the 2013 Confederations Cup known as the *Brazilian Spring*. The protests initially focused on increases in public transit costs, but eventually spread to include government corruption. By incorporating predictions from the underlying algorithms, we develop a method for incorporating social media information to make informed predictions particularly about these kinds of protests in which historical information is not sufficient for prediction.

4.2.2 Predictive Algorithm Alerts

A collection of four algorithms issues alerts of upcoming civil unrest events. While there is some overlap in the underlying data used by each algorithm, the algorithms have their own methodology for producing alerts. As such, certain algorithms may be better or worse predictors in various countries or regions. Each particular alert contains the location (country/state) and date

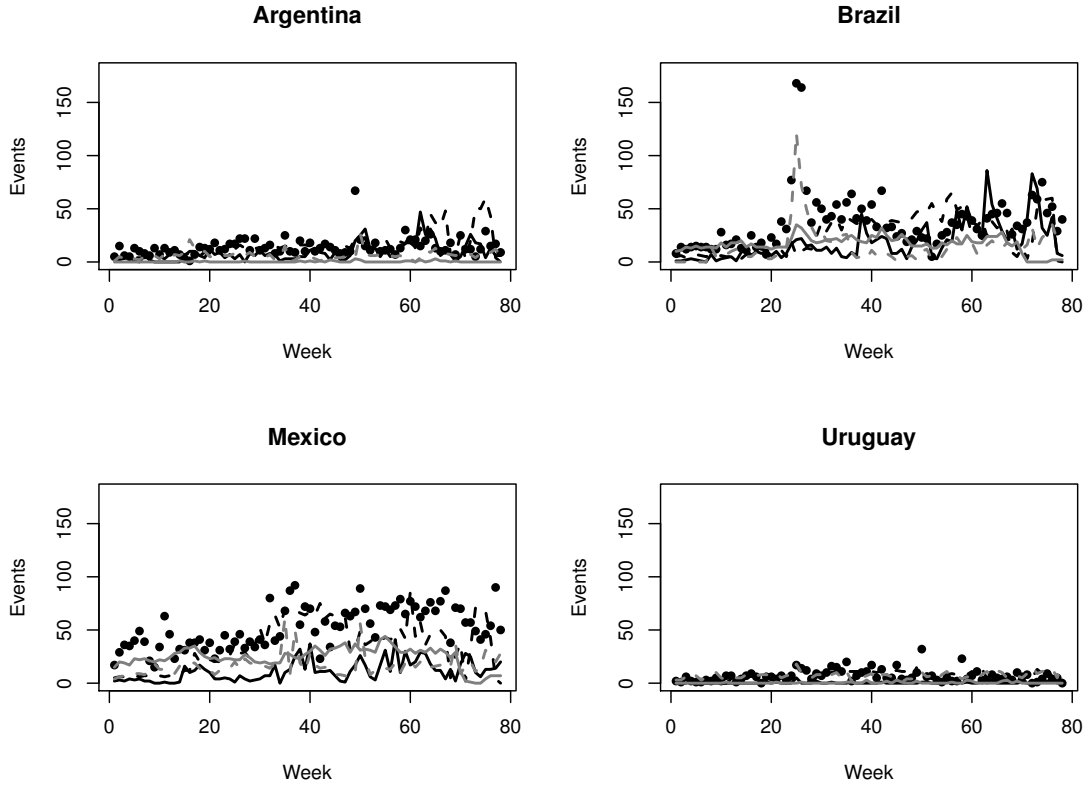


Figure 4.1: Event counts as black dots, with alert counts as lines (black - dash = TPP, black - solid = PP, gray - dash = DQE, gray - solid = KV)

(month/day/year) of a potential future protest. The four algorithms that issue discrete predictions are: Twitter Planned Protest (TPP), Dynamic Query Extraction (DQE), Planned Protest (PP), and Keyword Volume (KV). We provide a brief overview of the algorithms here, but additional information is available in Ramakrishnan et al. (2014). The TPP algorithm processes tweets searching for mention of future protest events. Similarly the PP algorithm is employed to search publicly available Facebook event pages and blogs for future protests. The KV algorithm uses counts of relevant protest related words (in Spanish and Portuguese) to predict future protests. While the KV algorithm contains a static set of keywords, the DQE algorithm dynamically learns a set of keywords associated with civil unrest, indicating future events. For each of these algorithms, alerts aggregated at the national level can also be seen in Figure 4.1 along with national event counts.

Additionally, each algorithm has a varying level of spatial resolution. For instance the KV algorithm focuses largely on major cities in each country while the PP algorithm is more widely distributed across countries. See Figure 4.2 for the spatial distribution of alerts for each algorithm in Mexico. From this figure we see that the TPP and PP algorithms have wider coverage across the

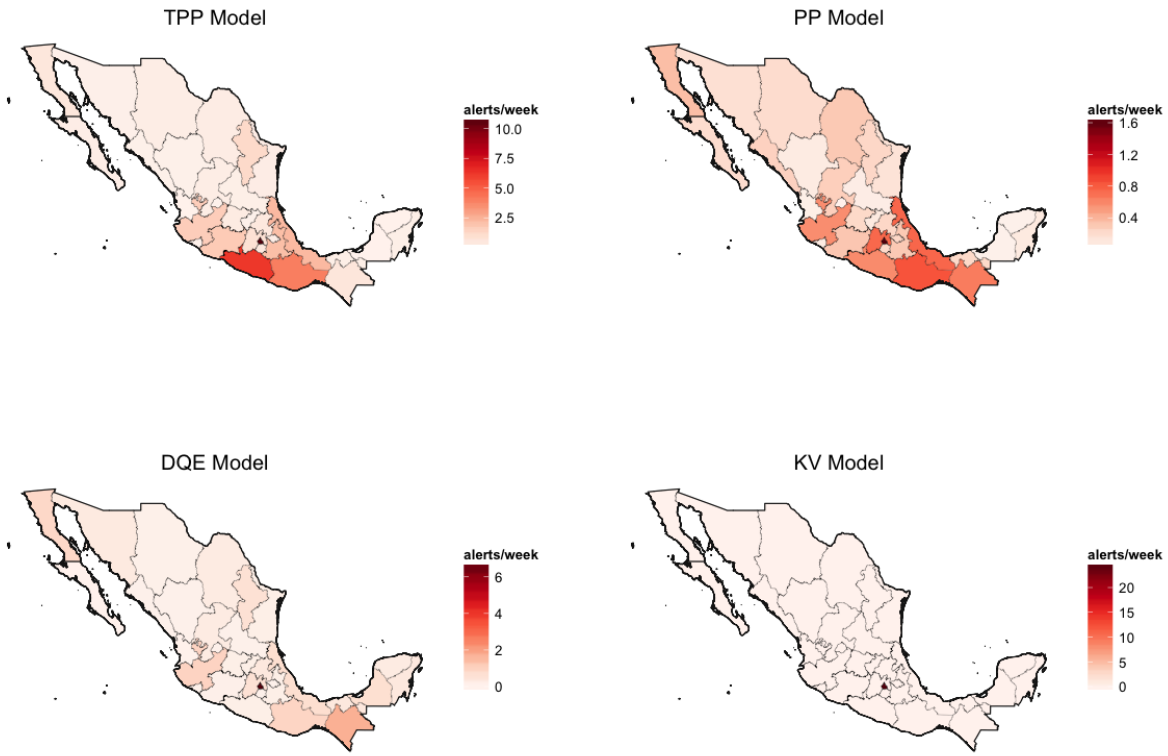


Figure 4.2: Spatial Resolution of Algorithms in Mexico

country. The DQE and KV algorithms issue most of their alerts in the capital district (Ciudad de México). These figures are fairly representative of the distribution of alerts in the other countries as well, as DQE and KV tend to issue alerts primarily in the capital cities as well as other large municipalities, while the TPP and PP issue alerts more widely across the country.

4.3 Multiscale Principles

The following subsections detail the multiscale partition for spatial data and multiscale factorization for Poisson data. Multiscale methods allow fitting richly specified spatiotemporal models with computational expediency achieved through parallelization from a coarse to fine decomposition of the data.

4.3.1 Multiscale Partition

Given data with an inherent hierarchical spatial structure (e.g. country/macroregion/state), the data are trivially multiscale. An example of a three level multiscale framework can be visualized as a multi-furcating tree as in Figure 4.3. Let Y_{tlj} denote the response for the j^{th} element of the l^{th} level

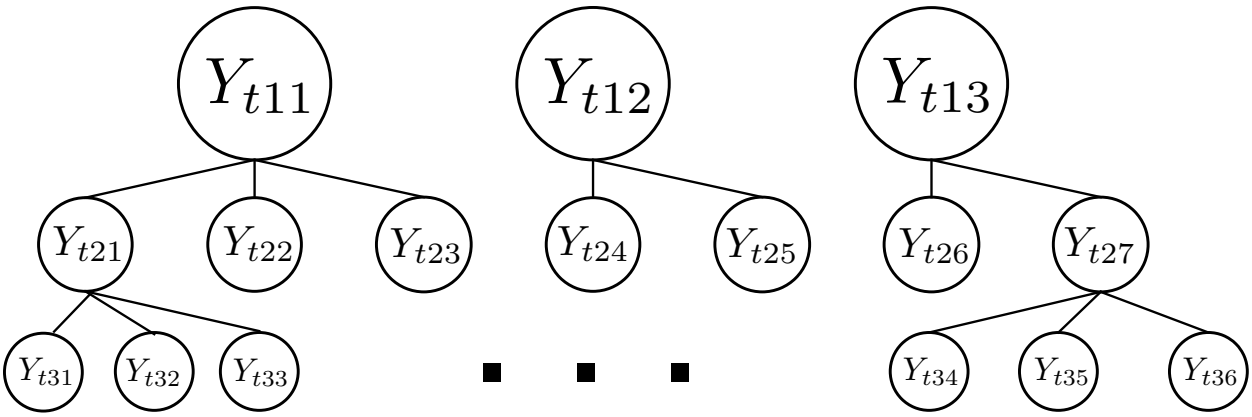


Figure 4.3: Example of three level multiscale structure

at time t, for instance Y_{t11} corresponds to the response for the $1st$ country at time t. Similarly Y_{t21} and Y_{t31} correspond to the first macroregion within the first country and the first state within the first country and macro region respectively.

There may be spatial effects present at different levels of resolution, some more global and others more local. With a multiscale framework, spatial effects are encoded by shared ancestors at different levels of the multiscale partition. Regional partitions are obtained such that socio-economic

factors and common sentiment make the spread of protests likely within the region. For instance, Figure 4.4 shows the regional partition of Mexico. In tree form, this partition would be represented

Figure 4.4: Multiscale partition for Mexico, where the six regions (from lightest to darkest) are Central Mexico, Northern Mexico, Pacific Coast, Yucatan, Baja, and the Bajio.

by a single node at the top, for Mexico, with six nodes below for the six regions. For the third level each region would have the number of states depicted in Figure 4.4. The appendix contains the complete list of cultural regions for each of the ten countries in the study.

4.3.2 Multiscale Factorization

With multiscale data, the multiscale factorization allows a decomposition of the joint likelihood enabling modular computation. The counts at the finest level are assumed to follow a Poisson distribution, $Y_{tLj}|\mu_{tLj} \sim \text{Poisson}(\mu_{tLj})$, where the count in the j^{th} location of the L^{th} level at time t is denoted Y_{tLj}. The factorization, originally shown in Kolaczyk and Huang (2001), is defined

as:

$$\prod_{j=1}^{n_L} p(Y_{tLj}|\mu_{tLj}) = \prod_{j=1}^{n_1} p(Y_{t1j}|\mu_{t1j}) \prod_{l=1}^{L-1} \prod_{j=1}^{n_l} p(\boldsymbol{Y}_{tD_{lj}}|Y_{tlj}, \boldsymbol{\omega}_{tlj}). \tag{4.1}$$

The likelihood can be factorized as the product of the coarsest level, $Y_{t1j}|\mu_{t1j} \sim \text{Poisson}(\mu_{t1j})$, and each level of the multiscale partition where the allocation to finer levels, $\boldsymbol{Y}_{tD_{lj}}|Y_{tlj}, \boldsymbol{\omega}_{tlj}$, is distributed according to a Multinomial$(Y_{tlj}, \boldsymbol{\omega}_{tlj})$. The counts for the descendants of region (lj) are contained in $\boldsymbol{Y}_{tD_{lj}}$ and the vector $\boldsymbol{\omega}_{tlj}$ contains the multiscale coefficients for the descendants of region (lj) at time t. The multiscale coefficients are the dynamic vector of probabilities for the multinomial distribution.

By implementing the multiscale partition, a conditional independence structure is induced between μ and $\boldsymbol{\omega}$ across all levels of resolution. The conditional independence does not imply a lack of spatial correlation, rather shared ancestors of the multiscale structure embed the spatial structure in the μ terms. While the temporal evolutions will be described in Sections 4.4.1 and 4.4.2, the key point with the multiscale factorization is that each of the time varying parameters from the Poisson and multinomial distributions are conditionally independent, allowing efficient parallel computation. This satisfies our first requirement of a scalable method for efficient computation of spatiotemporal data of increasing size.

4.4 Spatiotemporal Modeling

Fonseca and Ferreira (2016) have developed methods for multiscale spatiotemporal modeling of Poisson data; however, their method only permits covariates at the coarsest level. As a consequence, in their model, after applying the factorization in Equation (4.1), the covariates change the expected magnitude of the response at the coarse level. While including covariates at the coarse level is a desirable feature, it is also necessary to permit covariates at finer resolutions as well, which enable shifts in the allocation between subregions based on the distribution of alerts. Specifically, covariates at finer levels change the multiscale coefficients, and in turn the distribution between regions. Hence, we develop a novel model for including covariates at the coarse

level as well as each of the finer levels in a multiscale spatiotemporal Poisson framework. This provides a methodology for capturing the spatial structure across the hierarchies in the algorithm alerts satisfying the second requirement of our framework, incorporating hierarchical predictions.

4.4.1 Coarse Evolution

As shown in Equation (4.1) the response at the coarse level follows a Poisson distribution. Temporal modeling requires evolution in the mean terms for the Poisson distribution, which we state as follows:

$$\begin{aligned} Y_{t1j} &\sim \text{Poisson}(\mu_{t1j}) && (4.2)\\ \log(\mu_{t1j}) &= \eta_t + \boldsymbol{X}_{t1j}\boldsymbol{\beta}_{1j}, and && (4.3)\\ \eta_t &= \eta_{t-1} + \nu_t, \quad \nu_t \sim N(0, \tau_t^{-1}), && (4.4) \end{aligned}$$

where $\boldsymbol{X}_{t1j}$ are alerts aggregated for region $(1j)$. Specifically, $\boldsymbol{X}_{t1j} = [X_{t1j1}...X_{t1j4}]$, where X_{t1ji} is the log count of alerts, plus 1, for time t in region $(1j)$ from algorithm i. For prior specifications vague conjugate priors are recommended: $\tau_0 \sim Gamma(a, b)$, $\boldsymbol{\beta_{1j}} \sim N(\boldsymbol{m}, p^{-1}I)$, and $\eta_0 \sim N(m_\eta, p_\eta^{-1})$. Evolution of τ_t is modeled through a discount factor such that if $\tau_t|\mathcal{D}_t \sim Gamma(a_t, b_t)$ then $\tau_{t+1}|\mathcal{D}_t \sim Gamma(\delta a_t, \delta b_t)$, where $\mathcal{D}_t = \boldsymbol{Y}_{1:t,1j}$. The use of the discount factor preserves the expected precision, but results in increased variance by a factor of $1/\delta$. This model specification couples a latent time varying process η with information extracted from alerts to issue predictions. Additional details on computation and algorithmic implementation are provided in Section 4.5.

4.4.2 Fine Evolution

Again referring to Equation (4.1), allocation to the finer levels is computed via a multinomial distribution. To model the temporal evolution of the multiscale coefficients we introduce the following

framework:

$$\begin{aligned} \boldsymbol{Y}_{tD_{lj}} &\sim \text{Multinomial}(Y_{tlj}, \boldsymbol{\omega_{tlj}}) && (4.5)\\ \boldsymbol{\omega}_{tlj} &= \boldsymbol{\phi}_{tlj}/\mathbf{1}^T\boldsymbol{\phi}_{tlj} && (4.6)\\ \text{where} \quad \phi_{tlji} &= \exp(\gamma_{tlji} + \boldsymbol{X}_{tlji}\boldsymbol{\beta}_{lj}), \quad \text{and} && (4.7)\\ \boldsymbol{\gamma}_{tlj} &= \boldsymbol{\gamma}_{t-1,lj} + \boldsymbol{\epsilon}_{tlj}, \quad \boldsymbol{\epsilon}_{tlj} \sim N(0, \Sigma_t), && (4.8) \end{aligned}$$

where ϕ_{tlji} and γ_{tlji} correspond to the i^{th} element of the vectors $\boldsymbol{\phi}_{tlj}$ and $\boldsymbol{\gamma}_{tlj}$ respectively. Further, we impose $\mathbf{1}^T\boldsymbol{\gamma}_{tlj} = 0$ for identifiability. The evolution matrix Σ_t can have a general form, but we restrict it to $\Sigma_t = \tau_t^{-1}I$, where I is the identity matrix. The steps in Equations (4.6) and (4.7) can be thought of as a multinomial analogue to the logistic link function, this is known as the softmax function or an additive logistic transformation. Prior specification is similar to the coarse evolution with conjugate priors, where $\tau_0 \sim Gamma(a, b)$, $\boldsymbol{\beta_{lj}} \sim N(\boldsymbol{m}, p^{-1}I)$, and $\boldsymbol{\gamma}_0 \sim N(\boldsymbol{m}_\gamma, p_\gamma^{-1}I)$. Discounting is used for evolution of τ_t where if $\tau_t|\mathcal{D}_t \sim Gamma(a_t, b_t)$ then $\tau_{t+1}|\mathcal{D}_t \sim Gamma(\delta a_t, \delta b_t)$. Note that the parameters $\boldsymbol{\tau}$ and the values $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{m}$, and p are distinct between the fine evolution and coarse evolution models. Again, at the finer levels a latent time-varying process is coupled with information from our alerts to determine the multiscale coefficients which control the coarse-to-fine allocation across subregions.

4.4.3 Advantages of Multiscale Framework

A major advantage of the multiscale framework and the temporal evolution detailed here is the modular structure that is induced. Specifically, the set of mean terms for each coarse region, $\boldsymbol{\mu}_{1:T,1j}$, and set of multiscale coefficients for each fine partition, $\boldsymbol{\omega}_{1:T,lj}$ are conditionally independent *a posteriori* (Fonseca and Ferreira, 2016). Referring to the multiscale structure illustrated in Figure 4.3, each of the top nodes and every juncture in the tree can be computed in parallel providing scalability and computational efficiency. This is a sharp contrast to a dynamic GLM with Conditional AutoRegressive (CAR) term for spatial effects, where the random effects estimation would need to be conducted jointly. Hence, the multiscale structure is scalable for massive datasets

both in a spatial and temporal sense. Furthermore, the modular structure of the multiscale models also permit Sequential Monte Carlo (SMC) methods for streaming data or online predictions. This satisfies our first requirement of a scalable method. The multiscale framework also allows a natural way to incorporate hierarchical predictions, wherein predictions at the coarsest level effect the national counts and in contrast the finer level predictions shift the expected proportion of events across the region, satisfying the second requirement. Algorithm alerts are uncertain, particularly in a spatial sense as alerts issued for one state may actually better correspond to a neighboring state. Fortunately the multiscale model also implements a de facto error in variable approach, whereby predictions are smoothing within partitions satisfying the third requirement. Finally, the model is quite flexible and can easily handle modifications to the predictive algorithms or the addition of new predictive algorithms. No assumptions are made about the methodology of the underlying algorithms. If these algorithms were to be modified, the multiscale framework could be effortlessly refit to account for these changes satisfying the fourth requirement of our fusion framework, flexibility.

4.5 Computation

Given the modular framework of the multiscale model, the inference necessitates two procedures. One for the coarse level and the dynamic mean term of the Poisson evolution in Equations (4.2) - (4.4) and a second for the multinomial allocation and the dynamic vector of multiscale coefficients in Equations (4.5) - (4.8). As was mentioned earlier, each set of the Poisson means and multiscale coefficients can be independently estimated, which allows computation to be parallelized such that the entire run time would be equivalent to the evolution for one partition (assuming enough cores are available for computing).

We present a computational strategy based on SMC. Markov Chain Monte Carlo (MCMC) is typically the gold standard when fitting richly parameterized Bayesian models, particularly for retrospective analyses. However, if rapid estimation is the goal, for online estimation or the analysis of

streaming data, then SMC is preferable. Undertaking online 1-step ahead predictions with MCMC requires refitting the entire model at each time point. This is in stark contrast to a SMC approach in which the particles could easily be perturbed to construct a predictive distribution at the next time point. This results in runtime for each parallelized component is over 1000 times faster when implemented in SMC than MCMC. Hence, the SMC approach provides the rapid, scalable estimation of fully Bayesian models that we desire and cannot be achieved through MCMC.

4.5.1 SMC

SMC methods are popular for estimating state parameters in state-space models and provide a computational advantage when computing time is of the essence, particularly for prospective analyses conducted online. In contrast to MCMC where estimating a predictive distribution at time $t+1$ requires refitting the entire dataset, SMC uses the posterior at time t to construct a prediction at time $t+1$ in a sequential process. Specifically, consider the notation in Equations (4.2) - (4.4) and given a posterior for μ_t at time t, denoted as $p(\mu_t|\boldsymbol{Y}_{1:t})$, then SMC recursively solves the following equations at each successive time point $t+1$:

$$\text{Evolve state} \qquad P(\mu_{t+1}|\boldsymbol{Y}_{1:t}) = \int P(\mu_{t+1}|\mu_t)P(\mu_t|\boldsymbol{Y}_{1:t})d\mu_t \tag{4.9}$$

$$\text{Prediction for t+1} \qquad P(Y_{t+1}|\boldsymbol{Y}_{1:t}) = \int P(Y_{t+1}|\mu_{t+1})P(\mu_{t+1}|\boldsymbol{Y}_{1:t})d\mu_{t+1} \tag{4.10}$$

$$\text{Posterior for t+1} \qquad P(\mu_{t+1}|\boldsymbol{Y}_{1:t+1}) \propto P(Y_{t+1}|\mu_{t+1})P(\mu_{t+1}|\boldsymbol{Y}_{1:t}). \tag{4.11}$$

Hence, only incremental computations are required for a prospective analysis using SMC in contrast to an MCMC analysis in which the complete dataset would need to be used for each successive 1-step ahead prediction.

In cases where the above recursive equations do not have analytical solutions, a popular class of SMC methods for state-space models known as particle filters are commonly used. Whereas integration via MCMC methods is carried out by the iterations in the algorithm, particle filters estimate distributions and integrate functions by a discrete approximation based on a set of particles. For a comprehensive overview of particle based methods, see for instance Doucet et al. (2001). Several

variants of this method are Sequential Importance Sampling, Sequential Importance Resampling, Auxiliary Particle filters (Liu and Chen, 1998; Pitt and Shephard, 1999). However, one drawback to these methods is they do not provide a way to sample static parameters inside the SMC framework. Storvik (2002) and Liu and West (2001) have provided ways to handle this issue, utilizing sufficient statistics in Storvik and using mixture distributions in Liu and West (LW). Recently another approach detailed in Carvalho et al. (2010) known as Particle Learning (PL) has become popular. With PL sufficient statistics of static parameters are tracked as components in the particles. In certain cases where sufficient statistics are not available, such as the $\boldsymbol{\beta}$ values in our application, a mix of PL and the LW algorithms is used (Niemi, 2009; Dukic et al., 2012), whereby sufficient statistics are used when available and the LW framework otherwise. This SMC approach, which we implement in our application, is extremely fast computationally and provides all of the benefits of a fully Bayesian procedure.

Bayesian sequential analysis requires a prior distribution for the initial state of each parameter. For the coarse evolution model, a conjugate prior is used for the state vector, η_0, specifically $\eta_0 \sim N(log(\mu^*), \sigma^2)$, where μ^* is a historical average event count. A conjugate gamma prior is implemented for τ_0. To model dynamic evolution, a discount factor, $\delta \in (0, 1]$, is used to model time varying precision τ_t where if $\tau_t|\mathcal{D}_t \sim Gamma(a_t, b_t)$ then $\tau_{t+1}|\mathcal{D}_t \sim Gamma(\delta a_t, \delta b_t)$. The discount factor is set to 0.99 for all analyses in this article. Finally the prior for $\boldsymbol{\beta}$ is a vague normal prior. Our SMC algorithm uses PL and LW to estimate the state parameters as well as the static parameters; complete details are shown in the appendix, which also contain an MCMC implementation.

The algorithm for the fine evolution is quite similar to that of the coarse evolution and is presented in the appendix. Priors are needed for $\boldsymbol{\gamma}_0$, τ_0, and $\boldsymbol{\beta}$. Conjugate multivariate normal priors are specified for $\boldsymbol{\gamma}_0$. For identifiability $\boldsymbol{\gamma}_t$ is constrained to sum to zero. A conjugate prior from the gamma distribution is used for the precision term $\tau_0 \sim Gamma(a_0, b_0)$. The same discounting procedure as in the coarse evolution is used. A vague normal prior is specified for $\boldsymbol{\beta}$, which is sampled via a Metropolis within Gibbs step.

4.6 Results

We compare our novel multiscale framework using conditional Bayes' Factors (BF) with two competing models: a set of models run independently for each state and a multiscale model that only permits alerts at the top level. The independent models follow the specification in Equations (4.2) - (4.4).

To construct 1-step ahead predictions, we use the posterior predictive distribution to generate one week ahead predictions. The posterior predictive distribution at the finest level can be formulated as

$$p(Y_{t+1Lj}|\boldsymbol{Y}_{1:tLj}, \boldsymbol{X}_{1:t+1Lj}) = \int p(Y_{t+1Lj}|\boldsymbol{\Theta}, X_{t+1Lj}) p(\boldsymbol{\Theta}|\boldsymbol{Y}_{1:tLj}, \boldsymbol{X}_{1:tLj}) d\boldsymbol{\Theta},$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\eta}_{1:t1j}, \boldsymbol{\tau}_{1:t1j}, \boldsymbol{\beta}_{1j}, \boldsymbol{\gamma}_{1:t2j}, \boldsymbol{\beta}_{2j}, \boldsymbol{\Sigma}_{1:t2j}, ..., \boldsymbol{\gamma}_{1:tLj}, \boldsymbol{\beta}_{Lj}, \boldsymbol{\Sigma}_{1:tLj}\}$. The integration is conducted by sampling the particle distributions of our SMC algorithms.

A common Bayesian framework for model comparison is the use of BF, computed as

$$B_{12} = \frac{p(\mathcal{D}|\mathcal{M}_1)}{p(\mathcal{D}|\mathcal{M}_2)},$$

where $p(\mathcal{D}|\mathcal{M}_1)$ is the integrated likelihood of the data given model 1. Model evaluation via BF are summarized in Kass and Raftery (1995), but generally a log BF of 0 indicates little difference, log BF > 2 indicates positive evidence, and a log BF > 10 indicates strong evidence of an improvement in fit.

For spatiotemporal settings, conditional BFs are often used. In Vivar and Ferreira (2009), conditional Bayes factors are used for model selection. Let the joint predictive density under model 1 be $p_1(Y_{t^*+1}, ..., Y_T|\mathcal{D}_{t^*}) = \prod_{t=t^*+1}^{T} p_1(Y_t|\mathcal{D}_{t-1})$. Then given posterior samples an estimate of the predictive density is $\hat{p}_1(Y_{t+1}|\mathcal{D}_t) = \frac{1}{n}\sum_{i=1}^{N} p_1(Y_{t+1}|x_t^{(i)}, \theta^{(i)})$. This formulation allows the computation of conditional Bayes factors as:

$$B_{12} = \frac{\hat{p}_1(Y_{t*+1}, ..., Y_T|\mathcal{D}_{t^*})}{\hat{p}_2(Y_{t*+1}, ..., Y_T|\mathcal{D}_{t^*})}.$$

4.6.1 Spatiotemporal Multiscale Evaluation

The data consist of 78 weeks of civil unrest counts, the first twenty-five time points are used to initialize the models and then predictive distributions are computed for the remaining observations. Note that in the SMC framework, parameter estimates are learned sequentially so that predictions for time t include data up to time $t-1$. Similarly, the conditional BF computations use all information from time $t-1$ to predict time t. We compare the conditional log BF between our multiscale model and independent models as well as multiscale models that only permit covariates at the coarse level. Table 4.1 contains the log scale BF between the full multiscale model and the competing models.

Table 4.1: Log BF Table

Country	Multiscale Top	Independent
Argentina	13.5	-22.4
Brazil	6.2	40.6
Chile	4.1	1.0
Colombia	2.0	13.1
Ecuador	0.9	0.1
El Salvador	0.0	0.5
Mexico	7.9	14.1
Paraguay	4.6	6.2
Uruguay	1.0	-21.2
Venezuela	3.7	49.7
Total	43.9	81.7

On the whole this comparison gives very strong evidence to favor our multiscale specification in regards to the two competing models, with log BF of nearly 44 relative to a multiscale model only utilizing covariates at the country level and nearly 82 relative to the independent model, both are considerably larger than Kass's threshold of 10 for very strong evidence. To better understand the implications of our proposed approach, we take a closer look at the results.

The multiscale models incorporate spatial structure, whereas the independent models do not. This spatial sharing is beneficial in most countries; however, there does exist a set of protests events that are more localized. The best performance of the multiscale models occurs in Brazil and Venezuela. During this time period both countries experienced wide spread protesting: the Brazilian Spring in Brazil and the surge of violent protests across Venezuela in early 2014. These scenarios, where there is a cascading effect of protests would be expected to have the strongest spatial structure. Hence, it is not surprising that the performance of our multiscale model is the best in these countries. Incorporating the alerts at the finer levels is also beneficial as the predictions of our novel approach outperform existing multiscale methods. A more subtle issue related to sharing information spatially is that the alerts are uncertain in both space and time. The multiscale procedure implements a de facto error in variable approach by smoothing the effect of the alerts across commonalities in the multiscale structure. For example, an alert issued for Sao Paulo, Brazil may actually correspond to an instance of civil unrest in Rio de Janeiro, Brazil. The multiscale structure would account for this, whereas the independent models would not.

As was mentioned at the start of this paper, civil unrest is a difficult phenomenon to predict. In certain instances, such as the widespread national protests in Brazil and Venezuela protests cascade spatially and temporally. In other scenarios, protests may be localized events. For instance, consider week 49 in Argentina and note the large spike in Figure 4.1. These protests were focused largely in Córdoba, where the police force went on strike protesting stagnant wages. After widespread looting, police salaries were quickly increased and the protests dissipated. Before the protests had fully spread to neighboring provinces, the other provinces quickly followed suit by raising wages for police. The advantage of the independent models framework in Argentina is largely driven by this week. During this week, Córdoba experienced nineteen protests, whereas the second largest number of protests in a week was two. In this scenario the predictive algorithms are not particularly adept at predicting this set of protests. In fact, in Figure 4.1 many of the algorithms show a spike the week *after* the protests have subsided. Furthermore, the relationship between the alerts and events prior to week 49 indicates a weak signal. Hence, the predictive densities are quite similar for both models with an overwhelming majority of their mass much less

than the observed protest count. However, in this case the multiscale framework tends to have have slightly narrower predictive distributions given the spatial sharing of information. This results in less mass in the extreme tails of the distribution for the multiscale models and the penalties, under BF, for the extreme event counts are much less severe. In other countries, such as Ecuador and El Salvador the incorporation of alerts and spatial sharing via the multiscale models do little to improve predictions. These see very few protests and although the BF show a slight preference to the multiscale models, the differences are not substantial.

We emphasize that the best single model is our multiscale approach. However, it is important to realize that the set of independent models is a multiscale model with a different partition. Specifically, in this framework each state is contained in an independent partition. We are working on extensions to these methods which place a prior distribution on the partition itself, enabling model selection and averaging across partitions. However, these methods will be computationally complex and outside the realm of this work in which the emphasis is on, amongst other things, rapid computation for online predictions.

4.6.2 Prediction Trends & Maps

The predictions generated from our models enable visualizations of temporal trends and spatial maps. These visualizations can be created for each country in our study, but we focus our attention on Brazil. Brazil is noteworthy as the Brazilian Spring occurred during our study period. In addition to the Brazilian Spring moniker these protests are known by several names, including: the 2013 Confederations Cup riots, V for Vinegar Movement, and June Journeys. The peak in protests can been seen around weeks 24–26 in Figure 4.5. The protests initially started in response to increased public transportation fares, but later spread to include government corruption partly related to the massive federal spending on Confederations and World Cup stadiums and 2016 Summer Olympic venues. Figure 4.5 also displays the evolution of the multiscale coefficients. If the rise in protests was localized to a certain region, the dynamics of these would show a large spike in that region. As the distribution of the multiscale coefficients is relatively consistent during

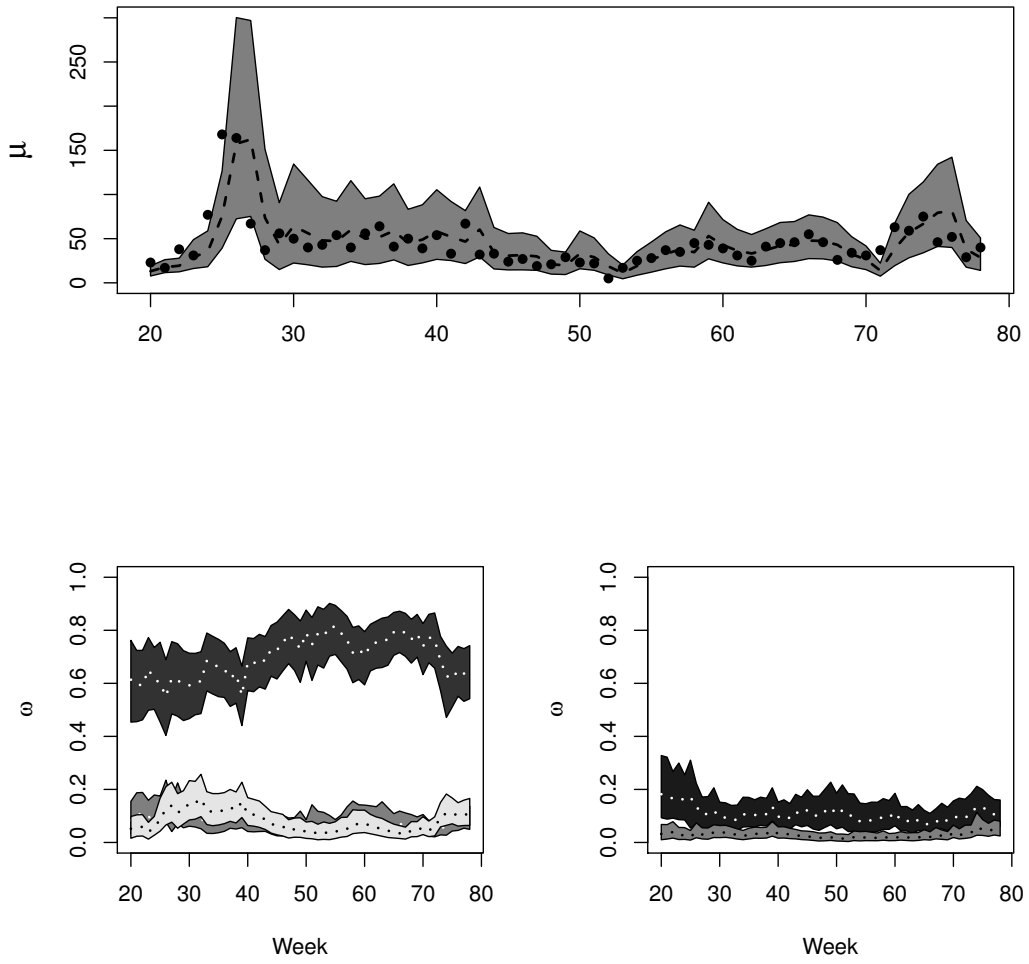


Figure 4.5: Prediction trends with credible intervals for Brazil. The top figure includes country level observed protest counts as dots. The bottom figures display the multiscale coefficients for regions in Brazil where two panels are used to show the five intervals. On the left figure, from dark to light are the Southeast, South, and Northeast regions. Similarly on the right figure from dark to light are the Central West and North regions.

this time period, we can infer that the increased level of protests are widely distributed nationally.

Incorporation of the predictive algorithms, particularly the DQE algorithm were helpful in predicting the increase in protests. The DQE algorithm learned a set of keywords relevant to the protests

by identifying the importance of the word *vinagre* through social media. *Vinagre* translates to vinegar which was used as a home remedy against tear gas and pepper spray from the police. In Figure 4.6, the 1-step ahead posterior predictive means are displayed for states in Brazil. The left figure is

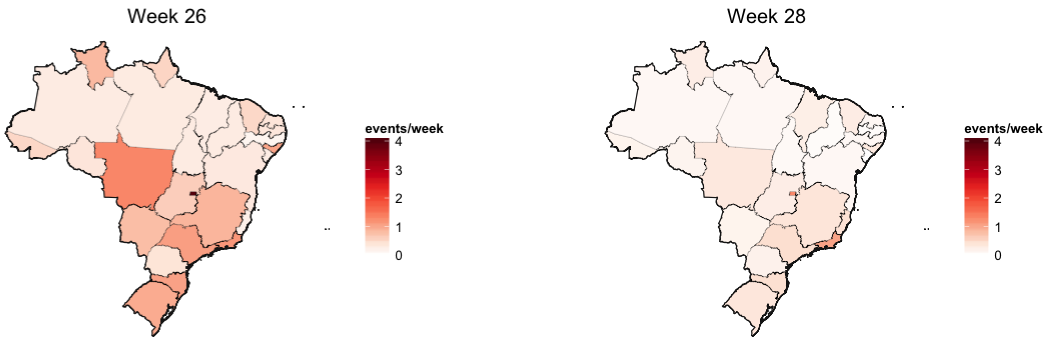


Figure 4.6: Posterior Predictive Mean for State Level Protests in Brazil, per million residents

for Week 26 at the height of the Brazilian Spring. The right figure shows the drastic decrease just two weeks later.

4.7 Discussion

With this work we detailed a novel multiscale spatiotemporal methodology for the analysis of areal count data. Our model satisfies the four guidelines outlined in Section 1: be scalable, incorporate hierarchical predictions, accommodate uncertainty in predictions, and be flexible. Based on BF this model also results in substantially better results than two competing methods. Civil unrest protests arise and spread in a complicated manner; however, our methods are able to predict trends and the spatial distribution of future protests.

On the whole our novel multiscale spatiotemporal model is a substantial improvement over competing methods. A key point is that multiscale factorization decomposes a complicated model into a set of smaller, simpler models. SMC does not work very well for large complex models,

but is useful for the smaller dimensional models induced by the multiscale factorization. Nevertheless, with future research we seek more flexible procedures. One option is to categorize and model protest separately. Intuitively, types of protests that tend to spread spatially could be handled in a different manner. Another option is to learn the multiscale partition. In our work, we have conditioned our models on the partition, but a prior could be placed on the partition providing a way to incorporate model uncertainty and choose most probable partitions. While these are both appealing, the computational considerations loom large, in particular for a procedure to learn the multiscale partition. A fully Bayesian approach would, almost certainly, require an intensive MCMC algorithm that would not meet our goal of a scalable, computationally efficient procedure.

Appendix A: SMC Algorithm Pseudocode

Algorithm 4 SMC Procedure for Coarse Evolution

procedure SMC COARSE
State $p(\gamma_0), p(\tau_0), p(\boldsymbol{\beta})$
Initialize particles
For (t in 1:T){
    Reweight particles $w_t^j \propto p(y_t|\gamma_{t-1}, \boldsymbol{\beta}_{t-1}, \tau_{t-1})$
    Approximate $p(\boldsymbol{\beta}_t|y_{1:t}) \approx N(\boldsymbol{\beta}_t; m_t, h^2 M_t)$, where $m_t = a\boldsymbol{\beta}_{t-1} + (1-a)\bar{\boldsymbol{\beta}}_{t-1}$ and $\bar{\boldsymbol{\beta}}_{t-1} = \sum_{j=1}^J w_t^j \boldsymbol{\beta}_{t-1}^j$ and $M_t = 1/J \sum_{j=1}^J w_t^j (\boldsymbol{\beta}_{t-1}^j - \bar{\boldsymbol{\beta}}_{t-1})(\boldsymbol{\beta}_{t-1}^j - \bar{\boldsymbol{\beta}}_{t-1})^T$
    Sample index $\boldsymbol{k}$ and draw $\boldsymbol{\beta}_t^j \sim N(m_t^k, h^2 M_t)$, **then for each k** {
        Propogate State $\gamma_t^j \sim p(\gamma_t|\gamma_{t-1}^k, \tau_t^k, \boldsymbol{\beta}_t^j)$
        Update sufficient statistics, $s_t^j = \mathcal{S}(s_{t-1}^k, \gamma_t^j, y_t, \boldsymbol{\beta}_t^j)$
        Sample auxiliary parameter, $\tau_t \sim p(\tau_t|s_t^j, \delta)$. } }
end procedure

Algorithm 5 SMC Procedure for Fine Evolution

procedure SMC FINE
State $p(\boldsymbol{\gamma}_0), p(\boldsymbol{\tau}_0), p(\boldsymbol{\beta})$
Initialize particles
For (t in 1:T){
    Reweight particles $w_t^j \propto p(y_t|\boldsymbol{\gamma}_{t-1}, \boldsymbol{\beta}_{t-1}, \boldsymbol{\tau}_{t-1})$
    Approximate $p(\boldsymbol{\beta}_t|y_{1:t}) \approx N(\boldsymbol{\beta}_t; m_t, h^2 M_t)$, where $m_t = a\boldsymbol{\beta}_{t-1} + (1-a)\bar{\boldsymbol{\beta}}_{t-1}$ and $\bar{\boldsymbol{\beta}}_{t-1} = \sum_{j=1}^{J} w_t^j \boldsymbol{\beta}_{t-1}^j$ and $M_t = 1/J \sum_{j=1}^{J} w_t^j (\boldsymbol{\beta}_{t-1}^j - \bar{\boldsymbol{\beta}}_{t-1})(\boldsymbol{\beta}_{t-1}^j - \bar{\boldsymbol{\beta}}_{t-1})^T$
    Sample index $\boldsymbol{k}$ and draw $\boldsymbol{\beta}_t^j \sim N(m_t^k, h^2 M_t)$ **then for each k** {
        Propogate State $\gamma_t^j \sim p(\boldsymbol{\gamma}_t|\boldsymbol{\gamma}_{t-1}^k, \boldsymbol{\tau}_t^j, \boldsymbol{\beta}_t^k)$
        Update sufficient statistics, $s_t^j = \mathcal{S}(s_{t-1}^k, \boldsymbol{\gamma}_t^j, y_t, \boldsymbol{\beta}_t^j)$
        Sample auxiliary parameter, $\Sigma_t^{-1} \sim p(\Sigma_t^{-1}|s_t^j, \delta)$. } }
end procedure

Appendix B: MCMC Framework for Spatiotemporal Multiscale Modeling

MCMC methods are frequently used for estimation of the state parameters in Bayesian state-space modeling. For normally distributed responses and evolution errors, analytical solutions are available under the well known Kalman Filtering equations (Kalman, 1960), provided that the error variances are known. When error variances are unknown the Kalman Filtering principles are embedded in an MCMC framework known as Forward-Filtering Backward Sampling (FFBS) (Frühwirth-Schnatter, 1994; Carter and Kohn, 1994). If the response is not normal, as in our case, closed form solutions are often not available for state parameters. This requires sampling methods and while traditional Metropolis-Hastings proposals can be applied, sampling large dimensional, highly correlated sets of parameters is problematic as the computing costs are high. A solution is to use Particle MCMC (P-MCMC) methods (Andrieu et al., 2010). P-MCMC uses a SMC technique to generate proposals for the state vector. In particular we use the Particle Gibbs (PG) algorithm, which allows the state vector to be sampled in a Gibbs sampler framework using the full conditional distributions. Tuning of the PG algorithm is controlled by the number of particles used in

the sampler.

Coarse Evolution

Bayesian sequential analysis requires a prior distribution for the initial state of each parameter. For the state vector, η_0, a conjugate prior is used, specifically $\eta_0 \sim N(log(\mu^*), \sigma^2)$, where μ^* is a historical average event count. A conjugate prior from the gamma distribution is used for the precision $\tau_0 = (\sigma_0^2)^{-1}$. A discount factor, $\delta \in (0, 1]$, is used to model time varying precision τ_t where if $\tau_t|\mathcal{D}_t \sim Gamma(a_t, b_t)$ then $\tau_{t+1}|\mathcal{D}_t \sim Gamma(\delta a_t, \delta b_t)$. The discount factor is set to 0.99 for all analyses in this article. Finally the prior for $\boldsymbol{\beta}$ is a vague normal prior.

Algorithm 6 details the sampling procedure. The PG algorithm is used for sampling $\boldsymbol{\eta}_{1:T}$ in which

Algorithm 6 P-MCMC Procedure for Coarse Evolution

1: **procedure** P-MCMC COARSE
2: State $p(\eta_0), p(\tau_0), p(\boldsymbol{\beta})$
3: Initialize $\boldsymbol{\eta}, \boldsymbol{\beta}, \boldsymbol{\tau}$
4: **For** (i in 1:Num.MCMC){
5: Sample $p(\boldsymbol{\beta}^{(i)}|\boldsymbol{\eta}^{(i-1)}, \boldsymbol{\tau}^{(i-1)})$ with MH proposal
6: Sample $p(\boldsymbol{\eta}^{(i)}|\boldsymbol{\beta}^{(i)}, \boldsymbol{\tau}^{(i-1)})$ with a conditional SMC update
7: Sample $p(\boldsymbol{\tau}^{(i)}|\boldsymbol{\beta}^{(i)}, \boldsymbol{\eta}^{(i)})$ from the full conditional, where $p(\tau_t|\tau_{1:t-1}, \boldsymbol{\eta}, \boldsymbol{\beta}) \sim Gamma(\delta a_{t-1}+1/2, \delta b_{t-1}+1/2(\eta_t - \eta_{t-1})^2)$. }
8: **end procedure**

the required tuning parameter is the number of particles. Depending on the size of T the number of particles can be adjusted accordingly to ensure proper mixing of $\boldsymbol{\eta}_{1:T}$. This is particularly important for the initial η values given the well known decay of particles under a bootstrap filter. The variance components $\boldsymbol{\tau}_{1:T}$ are sampled from the full conditional distribution and a Metropolis within Gibbs step is used to sample β.

Fine Evolution

The algorithm procedure for the fine evolution is quite similar to that of the coarse evolution and is presented in Algorithm 7. Priors are needed for $\boldsymbol{\gamma}_0$, Σ_0, and $\boldsymbol{\beta}$. Conjugate multivariate normal

Algorithm 7 P-MCMC Procedure for Fine Evolution

1: **procedure** P-MCMC FINE
2: State $p(\boldsymbol{\gamma}_0), p(\tau_0), p(\boldsymbol{\beta})$
3: Initialize $\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\tau}$
4: **For** (i in 1:Num.MCMC){
5: Sample $p(\boldsymbol{\beta}^{(i)}|\boldsymbol{\gamma}^{(i-1)}, \boldsymbol{\tau}^{(i-1)})$ with MH proposal
6: Sample $p(\boldsymbol{\gamma}^{(i)}|\boldsymbol{\beta}^{(i)}, \boldsymbol{\tau}^{(i-1)})$ with a conditional SMC update
7: Sample $p(\boldsymbol{\tau}^{(i)}|\boldsymbol{\beta}^{(i)}, \boldsymbol{\gamma}^{(i)})$ from the full conditional, where $p(\tau_t|\tau_{1:t-1}, \boldsymbol{\eta}, \boldsymbol{\beta}) \sim Gamma(\delta a_{t-1} + J/2, \delta b_{t-1} + 1/2(\boldsymbol{\gamma}_t - \boldsymbol{\gamma}_{t-1})^T(\boldsymbol{\gamma}_t - \boldsymbol{\gamma}_{t-1}))$. }
8: **end procedure**

priors are specified for $\boldsymbol{\gamma}_0$. For identifiability $\boldsymbol{\gamma}_t$ is constrained to sum to zero. We restrict $\Sigma_t = \tau_t^{-1} I$, where τ_t is a common precision term. A conjugate prior $\tau_0 \sim (\frac{1}{\tau_0^2})$ allows sampling from the full conditional distribution. A vague normal prior is specified for $\boldsymbol{\beta}$, which is sampled via a Metropolis within Gibbs step.

Appendix C: Multiscale Partitions

A complete list of the regional assignments of each state is available in Tables 4.2 and 4.3. The state = - and region = - pairs correspond to national level events that cannot be assigned to a spatial location.

Table 4.2: Regional Assignments (Part 1)

Country	Region	State	Country	Region	State
Argentina	-	-	Chile	Extreme South	Magellanes
Argentina	Andean Northwest	Catamarca	Chile	Northern Chile	Antofagasta
Argentina	Andean Northwest	Jujuy	Chile	Northern Chile	Arica y Parinacota
Argentina	Andean Northwest	La Rioja	Chile	Northern Chile	Atacama
Argentina	Andean Northwest	Salta	Chile	Northern Chile	Coquimbo
Argentina	Andean Northwest	Santiago del Estero	Chile	Northern Chile	Tarapacá
Argentina	Andean Northwest	Tucumán	Chile	Southern Chile	Araucanía
Argentina	Chaco	Chaco	Chile	Southern Chile	Bío Bío
Argentina	Chaco	Formosa	Chile	Southern Chile	Los Lagos
Argentina	Cuyo	Mendoza	Chile	Southern Chile	Los Ríos
Argentina	Cuyo	San Juan	Colombia	-	-
Argentina	Cuyo	San Luis	Colombia	Amazonia	Caquetá
Argentina	Pampas	Buenos Aires (district)	Colombia	Amazonia	Guaviere
Argentina	Pampas	Buenos Aires (province)	Colombia	Amazonia	Putumayo
Argentina	Pampas	Córdoba	Colombia	Andino	Bogotá
Argentina	Pampas	La Pampa	Colombia	Andino	Boyacá
Argentina	Pampas	Santa Fé	Colombia	Andino	Caldas
Argentina	Patagonia	Chubut	Colombia	Andino	César
Argentina	Patagonia	Neuquén	Colombia	Andino	Cundinamarca
Argentina	Patagonia	Río Negro	Colombia	Andino	Huila
Argentina	Patagonia	Santa Cruz	Colombia	Andino	Norte de Santander
Argentina	Tierra del Fuego	Tierra del Fuego	Colombia	Andino	Quindió
Brazil	-	-	Colombia	Andino	Risaralda
Brazil	Central West	Brasília	Colombia	Andino	Santander
Brazil	Central West	Goiás	Colombia	Andino	Tolima
Brazil	Central West	Mato Grosso	Colombia	Colombian Islands	San Andrés y Providencia
Brazil	Central West	Mato Grosso do Sul	Colombia	Costa Norte	Atlántico
Brazil	North	Acre	Colombia	Costa Norte	Bolívar
Brazil	North	Amapá	Colombia	Costa Norte	Córdoba
Brazil	North	Amazonas	Colombia	Costa Norte	La Guajira
Brazil	North	Pará	Colombia	Costa Norte	Magdalena
Brazil	North	Rondônia	Colombia	Costa Norte	Sucre
Brazil	North	Roraima	Colombia	Orinoquia	Arauca
Brazil	North	Tocantins	Colombia	Orinoquia	Casanare
Brazil	Northeast	Alagoas	Colombia	Orinoquia	Meta
Brazil	Northeast	Bahia	Colombia	Pacifica	Cauca
Brazil	Northeast	Ceará	Colombia	Pacifica	Chocó
Brazil	Northeast	Maranhão	Colombia	Pacifica	Nariño
Brazil	Northeast	Paraíba	Colombia	Pacifica	Valle del Cauca
Brazil	Northeast	Pernambuco	Ecuador	-	-
Brazil	Northeast	Piauí	Ecuador	Amazon Rainforest	Orellana
Brazil	Northeast	Rio Grande do Norte	Ecuador	Amazon Rainforest	Pastaza
Brazil	Northeast	Sergipe	Ecuador	Amazon Rainforest	Sucumbíos
Brazil	South	Paraná	Ecuador	Amazon Rainforest	Zamora Chinchipe
Brazil	South	Rio Grande do Sul	Ecuador	Andean Highlands	Azuay
Brazil	Southeast	Espírito Santo	Ecuador	Andean Highlands	Cañar
Brazil	Southeast	Minas Gerais	Ecuador	Andean Highlands	Carchi
Brazil	Southeast	Rio de Janeiro	Ecuador	Andean Highlands	Chimborazo
Brazil	Southeast	São Paulo	Ecuador	Andean Highlands	Cotopaxi
Chile	-	-	Ecuador	Andean Highlands	Imbabura
Chile	Central Chile	Libertador General Bernardo O'Higgins	Ecuador	Andean Highlands	Loja
Chile	Central Chile	Maule	Ecuador	Andean Highlands	Pichincha
Chile	Central Chile	Metropolitana	Ecuador	Andean Highlands	Santo Domingo de los Tsáchilas
Chile	Central Chile	Santiago	Ecuador	Andean Highlands	Tungurahua
Chile	Central Chile	Valparaíso	Ecuador	Coastal Lowlands	El Oro
Chile	Extreme South	Aisén	Ecuador	Coastal Lowlands	Esmeraldas

Table 4.3: Regional Assignments (Part 2)

Country	Region	State	Country	Region	State
Ecuador	Coastal Lowlands	Guayas	Paraguay	Northern Paranena	San Pedro
Ecuador	Coastal Lowlands	Los Ríos	Paraguay	Parana Plateau	Alto Paraná
Ecuador	Coastal Lowlands	Manabí	Paraguay	Parana Plateau	Caaguazú
Ecuador	Coastal Lowlands	Santa Elena	Paraguay	Parana Plateau	Caazapá
El Salvador	-	-	Paraguay	Parana Plateau	Canendiyú
El Salvador	Eastern El Salvador	Cabañas	Paraguay	Parana Plateau	Guairá
El Salvador	Eastern El Salvador	La Unión	Paraguay	Parana Plateau	Itapúa
El Salvador	Eastern El Salvador	Morazán	Paraguay	Southern Paranena	Asunción
El Salvador	Eastern El Salvador	San Miguel	Paraguay	Southern Paranena	Central
El Salvador	Eastern El Salvador	San Vicente	Paraguay	Southern Paranena	Cordillera
El Salvador	Eastern El Salvador	Usulután	Paraguay	Southern Paranena	Misiones
El Salvador	Western El Salvador	Ahuachapán	Paraguay	Southern Paranena	Ñeembucú
El Salvador	Western El Salvador	Chalatenango	Paraguay	Southern Paranena	Paraguarí
El Salvador	Western El Salvador	Cuscatlán	Uruguay	-	-
El Salvador	Western El Salvador	La Libertad	Uruguay	Atlantic Coast	Maldonando
El Salvador	Western El Salvador	La Paz	Uruguay	Atlantic Coast	Rocha
El Salvador	Western El Salvador	San Salvador	Uruguay	Central Interior	Cerro Largo
El Salvador	Western El Salvador	Santa Ana	Uruguay	Central Interior	Durazno
El Salvador	Western El Salvador	Sonsonate	Uruguay	Central Interior	Flores
Mexico	-	-	Uruguay	Northern Interior	Artigas
Mexico	Baja California	Baja California	Uruguay	Central Interior	Florida
Mexico	Baja California	Baja California Sur	Uruguay	Central Interior	Lavalleja
Mexico	Central Mexico	Ciudad de México	Uruguay	Central Interior	Treinta y Tres
Mexico	Central Mexico	Hidalgo	Uruguay	Northern Interior	Paysandú
Mexico	Central Mexico	México	Uruguay	Northern Interior	Rivera
Mexico	Central Mexico	Morelos	Uruguay	Northern Interior	Salto
Mexico	Central Mexico	Puebla	Uruguay	Northern Interior	Tacuarembó
Mexico	Central Mexico	Tlaxcala	Uruguay	Rio de la Plata	Canelones
Mexico	Central Mexico	Veracruz	Uruguay	Rio de la Plata	Colonia
Mexico	Central Mexico	Veracruz	Uruguay	Rio de la Plata	Montevideo
Mexico	Northern Mexico	Chihuahua	Uruguay	Rio de la Plata	Río Negro
Mexico	Northern Mexico	Coahuila	Uruguay	Rio de la Plata	San José
Mexico	Northern Mexico	Durango	Uruguay	Rio de la Plata	Soriano
Mexico	Northern Mexico	Nuevo León	Venezuela	-	-
Mexico	Northern Mexico	Sinaloa	Venezuela	Andes	Mérida
Mexico	Northern Mexico	Sonora	Venezuela	Andes	Táchira
Mexico	Northern Mexico	Tamaulipas	Venezuela	Andes	Trujillo
Mexico	Pacific Coast	Chiapas	Venezuela	Caribbean Islands	Nueva Esparta
Mexico	Pacific Coast	Colima	Venezuela	Central	Aragua
Mexico	Pacific Coast	Guerrero	Venezuela	Central	Carabobo
Mexico	Pacific Coast	Jalisco	Venezuela	Central	Caracas
Mexico	Pacific Coast	Michoacán	Venezuela	Central	Miranda
Mexico	Pacific Coast	Nayarit	Venezuela	Central	Vargas
Mexico	Pacific Coast	Oaxaca	Venezuela	Guayana	Amazonas
Mexico	The Bajio	Aguascalientes	Venezuela	Guayana	Bolívar
Mexico	The Bajio	Guanajuato	Venezuela	Guayana	Delta Amacuro
Mexico	The Bajio	Querétaro	Venezuela	Los Llanos	Apure
Mexico	The Bajio	San Luis Potosí	Venezuela	Los Llanos	Barinas
Mexico	The Bajio	Zacatecas	Venezuela	Los Llanos	Cojedes
Mexico	Yucatan	Campeche	Venezuela	Los Llanos	Guárico
Mexico	Yucatan	Quintana Roo	Venezuela	Los Llanos	Portuguesa
Mexico	Yucatan	Tabasco	Venezuela	Northeast	Anzoátegui
Mexico	Yucatan	Yucatan	Venezuela	Northeast	Monagas
Paraguay	-	-	Venezuela	Northeast	Sucre
Paraguay	Gran Chaco	Alto Paraguay	Venezuela	Northwest	Falcón
Paraguay	Gran Chaco	Boquerón	Venezuela	Northwest	Lara
Paraguay	Gran Chaco	Presidente Hayes	Venezuela	Northwest	Yaracuy
Paraguay	Northern Paranena	Amambay	Venezuela	Northwest	Zulia
Paraguay	Northern Paranena	Concepción			

Bibliography

Andrieu, C., Doucet, A., and Holenstein, R. (2010), "Particle Markov chain Monte Carlo cethods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72, 269–342.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008), "Gaussian predictive process models for large spatial data sets," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 825–848.

Banks, D., Datta, G., Karr, A., Lynch, J., Niemi, J., and Vera, F. (2012), "Bayesian CAR models for syndromic surveillance on multiple data streams: Theory and practice," *Information Fusion*, 13, 105 – 116, special Issue on Information Fusion Applications to Human Health and Safety.

Besag, J. (1974), "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 192–236.

Bliznyuk, N., Paciorek, C. J., Schwartz, J., and Coull, B. (2014), "Nonlinear predictive latent process models for integrating spatio-temporal exposure data from multiple sources," *The Annals of Applied Statistics*, 8, 1538–1560.

Brodley, C. E. (1993), "Addressing the selective superiority problem: Automatic algorithm/model class selection," in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 17–24.

Carter, C. K. and Kohn, R. (1994), "On Gibbs sampling for state space models," *Biometrika*, 81, pp. 541–553.

Carvalho, C. M., Johannes, M. S., Lopes, H. F., and Polson, N. G. (2010), "Particle learning and smoothing," *Statistical Science*, 25, 88–106.

Doucet, A., De Freitas, N., and Gordon, N. (2001), *An Introduction to Sequential Monte Carlo Methods*, Springer.

Dukic, V., Lopes, H. F., and Polson, N. G. (2012), "Tracking epidemics with Google flu trends data and a state-space SEIR model," *Journal of the American Statistical Association*, 107, 1410–1426.

Ferreira, M. A. R., Bertolde, A. I., and Holan, S. H. (2010), "Analysis of economic data with multiscale spatio-temporal models." *In Handbook of Applied Bayesian Analysis, eds. A. O'Hagan and M. West*.

Ferreira, M. A. R., Holan, S. H., and Bertolde, A. I. (2011), "Dynamic multiscale spatiotemporal models for Gaussian areal data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73, 663–688.

Ferreira, M. A. R. and Lee, H. K. H. (2007), *Multiscale modeling: a Bayesian perspective*, Springer.

Fonseca, T. C. O. and Ferreira, M. A. R. (2016), "Dynamic multiscale spatiotemporal models for Poisson data," *Journal of the American Statistical Association*, to appear.

Frühwirth-Schnatter, S. (1994), "Data augmentation and dynamic linear models," *Journal of Time Series Analysis*, 15, 183–202.

Furrer, R., Genton, M. G., and Nychka, D. (2006), "Covariance tapering for interpolation of large spatial datasets," *Journal of Computational and Graphical Statistics*, 15.

Gramacy, R. B. and Lee, H. K. (2008), "Bayesian treed Gaussian process models with an application to computer modeling," *Journal of the American Statistical Association*, 103, 1119–1130.

Kalman, R. E. (1960), "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, 82, 35–45.

Kass, R. E. and Raftery, A. E. (1995), "Bayes factors," *Journal of the American Statistical Association*, 90, 773–795.

Kolaczyk, E. D. (1999), "Bayesian multiscale models for Poisson processes," *Journal of the American Statistical Association*, 94, 920–933.

Kolaczyk, E. D. and Huang, H. (2001), "Multiscale statistical models for hierarchical spatial aggregation," *Geographic Analysis*, 33, 95–118.

Liu, J. and West, M. (2001), "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo methods in practice*, Springer, pp. 197–223.

Liu, J. S. and Chen, R. (1998), "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, 93, 1032–1044.

Niemi, J. (2009), "Bayesian analysis and computational methods for Dynamic Modeling," Ph.D. thesis, Duke University.

Pitt, M. K. and Shephard, N. (1999), "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, 94, pp. 590–599.

Ramakrishnan, N., Butler, P., Muthiah, S., Self, N., Khandpur, R., Saraf, P., Wang, W., Cadena, J., Vullikanti, A., Korkmaz, G., Kuhlman, C., Marathe, A., Zhao, L., Hua, T., Chen, F., Lu, C. T., Huang, B., Srinivasan, A., Trinh, K., Getoor, L., Katz, G., Doyle, A., Ackermann, C., Zavorin, I., Ford, J., Summers, K., Fayed, Y., Arredondo, J., Gupta, D., and Mares, D. (2014), "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*, New York, NY, USA: ACM, KDD '14, pp. 1799–1808.

Storvik, G. (2002), "Particle filters for state-space models with the presence of unknown static parameters," *IEEE Transactions on Signal Processing*, 50, 281–289.

Vivar, J. C. and Ferreira, M. A. R. (2009), "Spatiotemporal models for Gaussian areal data," *Journal of Computational and Graphical Statistics*, 18, 658–674.

Wolpert, D. H. (1992), "Stacked generalization," *Neural Networks*, 5, 241–259.

Chapter 5

Discussion

In the introduction we have introduced the concept of model fusion and discussed related statistical literature. Chapter 2 outlined theoretical properties for model fusion, presented an algorithm for prediction, and showed results for predicting the occurrence of civil unrest in six capital cities across Central and South America. Chapter 2 will be submitted to a journal in concert with the completion of this dissertation. In the context of an applied setting, Chapter 3 uses a simplified version of algorithm detailed in Chapter 2, to combine model output with the intent of optimizing similarity between predicted and observed cases of civil unrest with constraints on precision and recall. Chapter 3 has been accepted and published in *Technometrics*. In Chapter 4 we turned our attention to model fusion in a spatiotemporal setting. This required developing a novel spatiotemporal multiscale framework to provide scalable, efficient computing while incorporating hierarchical alerts with layers of uncertainty. This framework was shown to be superior to competing models. Chapter 4 has been accepted in *JRSS: Series C(Applied Statistics)*.

While this dissertation ends, extensions and future work abound. In particular, there are three concepts related to the work presented here that are currently underway. First, a multivariate extension of the method presented in Chapter 4 is being developed. This will be useful for multivariate counts, such as the inclusion of information pertaining to the type of protest and the population protesting for the data in Chapter 4. A second idea involves an algorithm for adaptively learning

the multiscale structure. A prior controlling the partitions of the multiscale can be introduced, which will enable selection of maximum *a posteriori* structures or averaging across several structures. Finally the multiscale framework, likely in concert with the partitioning prior, can be used for point processes in addition to areal data.