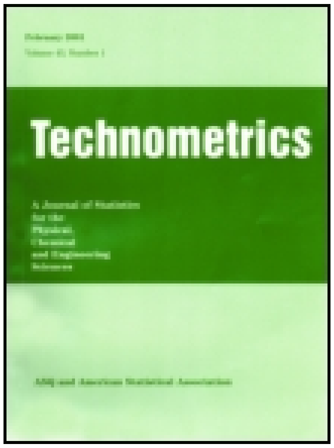


Technometrics
Publication details, including instructions for authors and subscription information:
http://amstat.tandfonline.com/loi/utch20

Bayesian Model Fusion for Forecasting Civil Unrest

Andrew Hoegh[ab], Scotland Leman[ab], Parang Saraf[bc] & Naren Ramakrishnan[bc]
[a] Department of Statistics, Virginia Tech
[b] Discovery Analytics Center, Virginia Tech
[c] Department of Computer Science, Virginia Tech

Accepted author version posted online: 13 Feb 2015.

To cite this article: Andrew Hoegh, Scotland Leman, Parang Saraf & Naren Ramakrishnan (2015): Bayesian Model Fusion for Forecasting Civil Unrest, Technometrics, DOI: 10.1080/00401706.2014.1001522

To link to this article: http://dx.doi.org/10.1080/00401706.2014.1001522

Bayesian Model Fusion for Forecasting Civil Unrest

Andrew Hoegh[1,2], Scotland Leman [1,2], Parang Saraf[2,3], and Naren Ramakrishnan[2,3]

[1]Department of Statistics, Virginia Tech
[2]Discovery Analytics Center, Virginia Tech
[3]Department of Computer Science, Virginia Tech

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

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 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 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 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 agnostic 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 2 compares fusion with a single super-model and details the underlying models that are inputs to the fusion mechanism. Section 3 defines civil unrest for the purposes of this analysis and details the structure of the data including the alert-event matching scheme. Section 4 describes the Bayesian model fusion algorithm implemented to model civil unrest. Section 5 provides an overview of the case study for predicting civil unrest in Argentina, Brazil, and Mexico. Section 6 concludes with a discussion.

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.

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 PRELIMINARIES

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.

3.2 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 2.

For instance, in the second alert-event pair the 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).

3.2.1 Alert-Event Matching Illustration

Consider the visual depiction of the process as shown in Figure 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 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.

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.

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.

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 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.

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:

$$\eta = \frac{c_{01}}{c_{10} + c_{01}}. \tag{1}$$

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.

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:

$$E = 1: \qquad \mathbf{Z_1} \sim N(\boldsymbol{\mu}_1, \mathbf{R}_1) \tag{2}$$

$$E = 0: \qquad \mathbf{Z_0} \sim N(\boldsymbol{\mu}_0, \mathbf{R}_0) \tag{3}$$

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,

$$P[\boldsymbol{m}|E = i, \boldsymbol{\mu}_i, \boldsymbol{R}_i] \quad = \quad \int_{Z^*} (2\pi)^{-mn/2} |\boldsymbol{R}_i|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{Z} - \boldsymbol{\mu}_i)' \boldsymbol{R}_i^{-1} (\mathbf{Z} - \boldsymbol{\mu}_i)) d\boldsymbol{Z}, \tag{4}$$

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 (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.

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 (1). This posterior distribution can be computed as

$$P[E = 1|\boldsymbol{m}] = \int \frac{P[\boldsymbol{m}|E = 1, \boldsymbol{\mu}_1, \boldsymbol{R}_1]P[E = 1]}{P[\boldsymbol{m}|E = 1, \boldsymbol{\mu}_1, \boldsymbol{R}_1]P[E = 1] + P[\boldsymbol{m}|E = 0, \boldsymbol{\mu}_0, \boldsymbol{R}_0]P[E = 0]} p(\boldsymbol{\Theta}) d\boldsymbol{\Theta}, \tag{5}$$

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.

5 BAYESIAN MODEL FUSION FOR CIVIL UNREST

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.

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, (2) and (3), can be seen in Table 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 type, $P[\boldsymbol{m}|E = i, \boldsymbol{\mu}, \boldsymbol{R}]$, is computed via (4). The output of the MCMC procedure necessary for fusion decisions are the posterior probabilities $P[E = 1|\boldsymbol{m}]$. Table 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 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.

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, 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 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 4.

Using the 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 4.

With a large η, fusion 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 5.

By suppressing the lowest quality alerts at each tuning 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 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.

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

   (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 (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 (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 $\boldsymbol{m} \in \mathcal{M}$.

References

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 new' 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.

Table 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}

Table 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)

Table 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

Table 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	$\eta = 0.80$

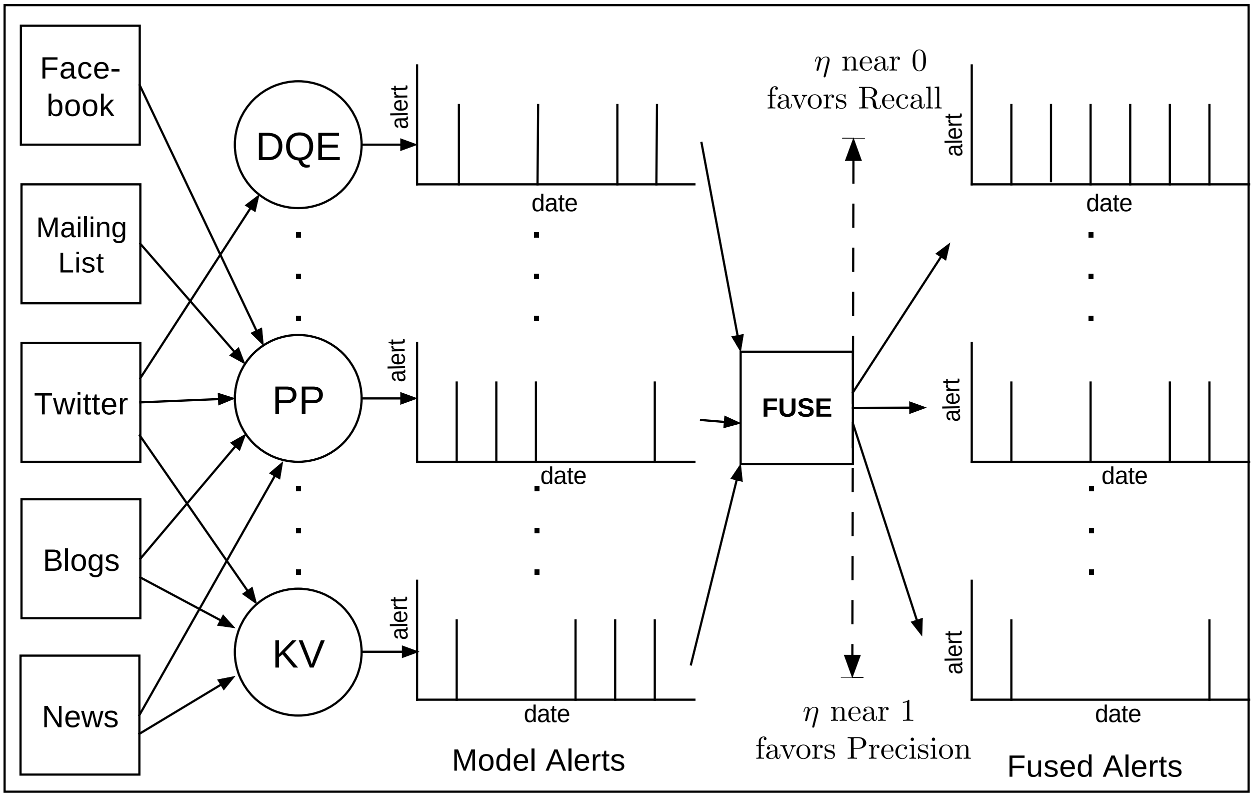
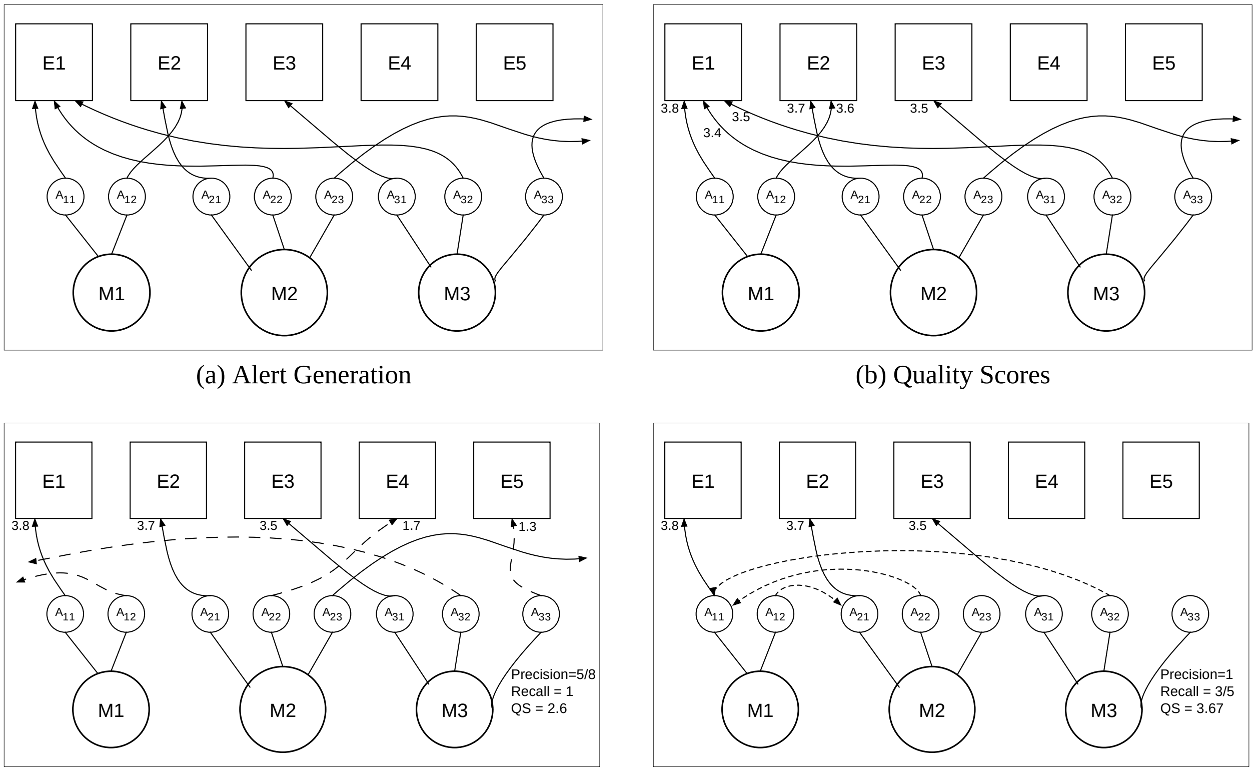


Figure 1: *System Informatics* framework for Bayesian model fusion, where DQE, PP, and KV are underlying models detailed in Sec 2.1 .

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 2: Matched alert-event pairs



(a) Alert Generation

(b) Quality Scores

(c) Matching without Fusion

(d) Matching with Fusion

Figure 3: Overview of alert-event matching.

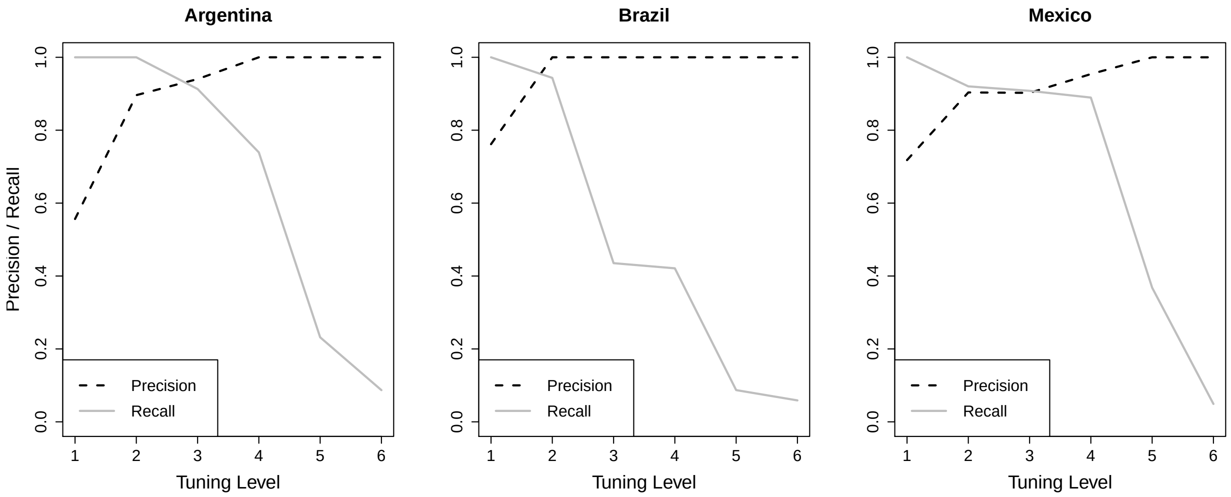


Figure 4: Precision Recall tradeoff by the tuning levels defined in Table 4

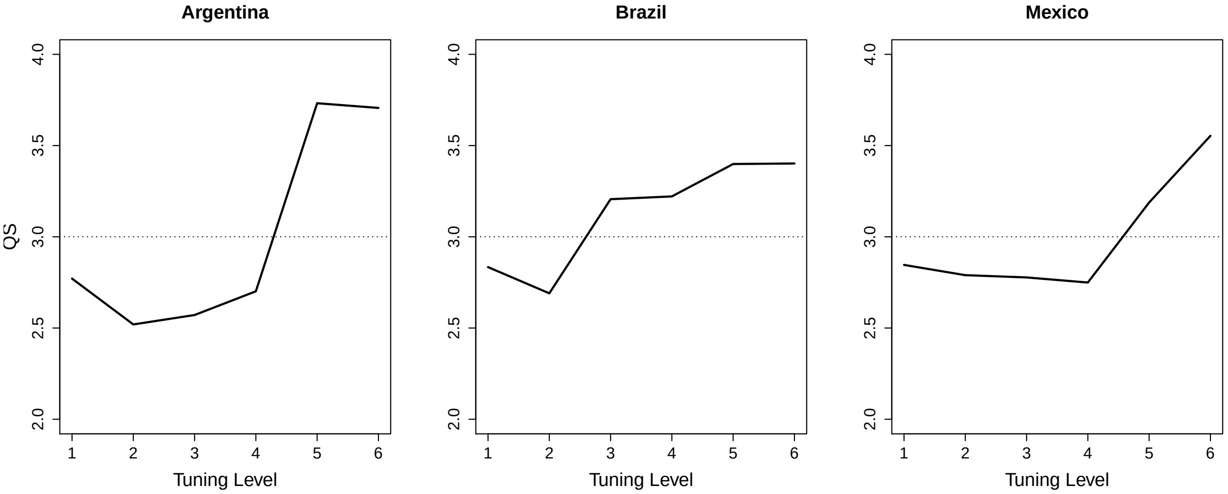


Figure 5: Mean QS in solid line, dashed line represents target QS of 3.0.

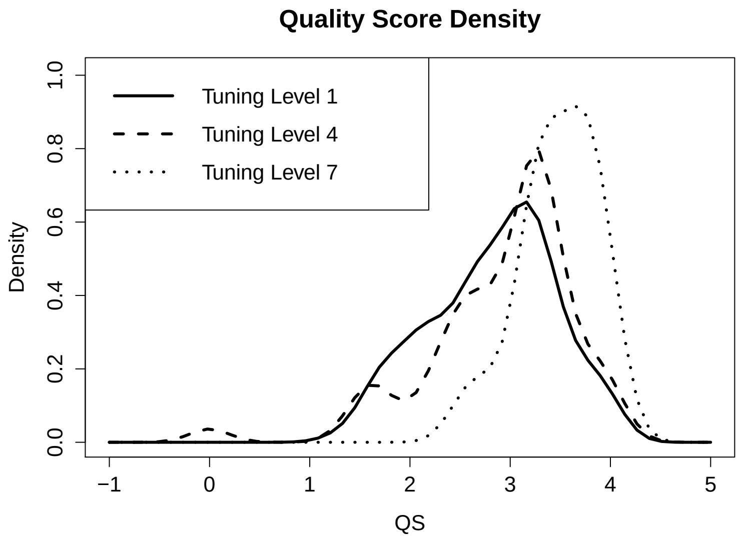


Figure 6: KDE of mean QS combined for all three countries for three specified tuning levels.