# Predicting Mass Movements with Public Data

Brian Higgins, Justin Lai

***Abstract*—The Internet has become a hotbed of social activism and political organization. We have created a classifier meant to predict when a mass movement will occur in a given metropolitan area. This will be a useful tool for activists, regulators, police and bystanders, by allowing them to prepare for mass movements or civil unrest. We have leveraged Google Trends data, employment data, and market sentiments to build our predictor. Given a region's recent search data, and the overall national sentiment, what is the likelihood that people will organize a large protest? We will discuss our methods, data, motivation and conclusions later, but our model has achieved a precision of over 80% while maintaining an accuracy and recall over 75%. Which is a useful predictor of whether a mass movement will occur in a specific metropolitan area.**

## I. Introduction

THE ability for large groups of people to communicate instantaneously has completely changed the world. In the last 30 years, the rise of the internet has enabled all sorts of collaboration, exchange of ideas, and the rapid dissemination of news to the population. With the ability to learn, communicate and plan across any distance, or without delay, large institutions of the world are being held to a higher standard than ever before. When our institutions fail, that leads to anger, social unrest, and the desire to make change.

We have seen the power of grass roots uprisings, protests, and mass movements over and over through history. Whether it is the civil rights movement in America, the destruction of Apartheid in South Africa or the Arab Spring in the Middle East, people's ability to communicate and organize is only improving with technology. But while mass movements can be incredible ways to make positive change, they are not always peaceful, safe and productive. Many protests lead to mass destruction of property, theft, and even violence.

Our goal is simple, we have built resources to help predict mass movements. Not out of a desire to prevent people from enacting change, but rather to allow everyone to be aware of their neighbor's intent. To give everyone the ability to speak out and join a movement, or avoid an unneeded inconvenience or confrontation.

We will input the rates at which people are searching given search terms, data over the financial market, financial market sentiment, and unemployment rates. We will output a label for whether or not a protest will happen in that region in the next 2 weeks.

## II. Related Work

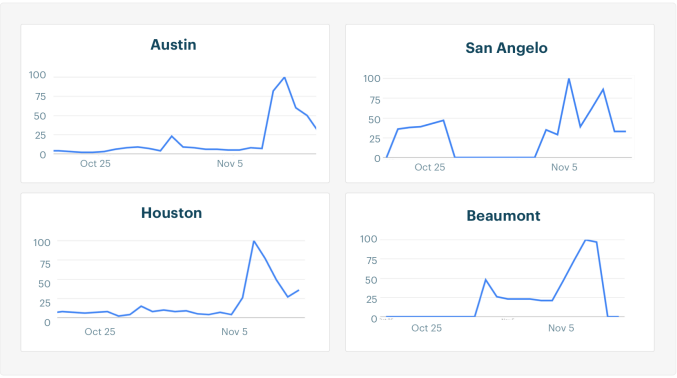
A primary source of inspiration for this project came from 'Predicting Crowd Behavior with Big Data' by Nathan Kallus of MIT. His work used natural language processing to analyze news data and predict major protests in the Middle East, with a focus on the coup d'etat in Egypt. Kallus was able to achieve a fairly accurate predictor for the events in Egypt.

Another source of inspiration was "Machine Learning and Conflict Prediction: A Use Case" by Chris Perry, which investigates the use of machine learning to predict conflicts in Africa. The potential accuracy of the project may have been limited due to the sheer geographic scope of the project. Towards the end of his paper, Perry notes the potential usefulness of Google's GDelt Events database in providing more accurate predictions. With this in mind, we decided to query data from the GDelt database through Google BigQuery.

## III. Data Set and Features

We mentioned earlier the possibly ethical issues to using Twitter data. Instead, we used Google Trends keyword data, as well as market sentiment and unemployment data from Quandl.

Google Trends provides scores of 1-100 for each keyword queried for a specific geographic location; the geographic locations can specify countries, states/counties, or metropolitan areas. The score is scaled between 1-100, and is calculated based on two metrics: (1) the popularity of the search term in a geographic region, relative to total searches in that region, and (2) the relative popularity of the search term compared to its relative popularity in other regions. For example,




[

Fig. 1. Google Trend scores for the search term 'protest', queried from 10/25/16 to 11/15/16 in various Texan metro areas. Austin and Houston held major protest rallies post-election, whereas Beaumont and San Angelo did not.

An initial analysis of Google Trends data indicated its potential usefulness in predicting protests. In Figure 1, the keyword queried is 'protest', and the graphs indicate the relative change in popularity of the keyword from Oct 25 - Nov 15, 2016. The two graphs on the left represent Austin and Houston, whereas the graphs on the right indicate San Angelo and Beaumont. We know that major post-election protests occurred in Austin and Houston, but not in San Angelo and Beaumont. The Google

Trends data for our date range shows significant spikes or relative search prevalence in Austin and Houston. However, in San Angelo and Beaumont, these spikes are much less prominent. In Beaumont, the Trends score hovers around 25 for roughly the week preceding the election. This indicates that post-election, the volume of searches for 'protest' was not as relatively post-election as it was in Houston and Austin. This coincides with the relative levels of post-election protest activity in the four metropolitan areas.

We identified 30 major protests in 26 metropolitan areas in the United States and Canada between Mar 5, 2011 to Dec 1, 2016. These included the Occupy Wall Street protests beginning on Sept 17, 2011, the Vancouver Hockey Riots beginning on June 5, 2011, the St. Louis Black Lives Matter protests beginning on Aug 30, 2016. For each query, the start date began three months prior to the start of the protest date, and ended on the first day of protests. For example, the query for the Vancouver riots sourced data from Mar 5 - Jun 5, 2011. Each query contained 30 keywords related to protests. In total, this resulted in around 81,000 data points.

We also sourced market sentiment, market data, and unemployment data from Quandl. These features were weekly, and were sourced from Mar 5, 2011 to Dec 1, 2016. Market sentiment gives a percentage of investors who express specific sentiments about the market (bullish or bearish). We also included weekly S&P 500 highs and lows, as well as weekly unemployment rates as features. While these statistics are not sourced by metro area like Google Trends, they often correlate with the sociopolitical unrest of local metropolitan areas.

*Note on project ethics*

We originally submitted a request to Twitter in order to use the GNIP Enterprise API for data sourcing. However, Twitter denied our request. Twitter specified that the project could be used for potentially unethical reasons related to surveillance or to infringe upon the rights of Twitter users. With this in mind, and at Twitter's request, we avoided using Twitter data through official and unofficial APIs.

## IV. Methods

We applied several learning algorithms to our problem with a mixture of results. We applied Support Vector Machine with Radial Basis Function Kernel, Bernoulli Naive Bayes, Logistic Regression, and Gradient Boosting to our binary classification data. And, for multi-class classification, we used SVM-RBF, Logistic Regression, and Gradient Boosting.

We will now discuss each of these algorithms chosen, why they were chosen, and what assumptions and benefits, and issues they bring to our problem statement.

**SVM with RBF Kernel**:

Support vector machines are a very common way to address classification problems in supervised learning. SVM's optimize hinge loss, which is defined as:

$$\nabla Loss(K^{(i)}, \vec{y}^{(i)}) = \begin{cases} -\vec{y}^{(i)} K^{(i)} & \text{if } \alpha \vec{y}^{(i)} K^{(i)} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

. This equation for loss intuitively makes sense, because when $\vec{y}^{(i)}$ and $K^{(i)}$ share the same sign they will be positive. Also, as our confidence in a prediction grows, so does K. Therefore, we do not update our loss function when we predict correctly with any reasonable confidence.

**Naive Bayes**:

Naive Bayes makes the strongest assumptions of any of the algorithms we chose to test. Naive Bayes assumes all data points are completly independant, which allows joint probabilities to be modeled as products. As products, they can be expanded to conditional probabilites using bayes rule. Additionally, it is a generative algorithm, which means it models $p(x \mid y)$ for each possible choice of y, then classifies with the most likely option.

So when we make predictions, we are trying to solve the probability: $p(y = 1 \mid x) = \frac{p(y=1)p(x|y=1)}{p(x)}$. And since we modeled $p(x \mid y)$ in our learning, we can make a prediction for that probability.

**Logistic Regression**

Logistic regression is a bread and butter classification algorithm, and the first we discussed in CS229. It is a discriminatory algorithm, and is used for classification because it is smooth over the range (0, 1).

It classifies using the sigmoid function:

$$h_\theta(X) = \frac{1}{1 - e^{-\theta^T X}} \quad (2)$$

and trains using gradient Ascent because there is no closed form for the likelihood:

$$l(\theta) = \sum_{i=1}^{m} y^{(i)} log\left(h_\theta(x^{(i)})\right) + \left(1 - y^{(i)}\right) log\left(1 - h_\theta(x^{(i)})\right) \quad (3)$$

**Gradient Boosting**:

Boosting defines a set of algorithms meant to combine a series of weak learners into a good classifier for a data set. While boosting algorithms generally fit data very closely, they tend to over fit data because of the high dimensional nature of combining a series of classifiers into one.

The idea with this algorithm is to apply the best weak learner into the classier, and combine it with the previous classifier. By doing this repeatedly, we can build up a classifier to fit the data very tightly. In this way, we apply coordinate descent, which is basically gradient descent on a single given feature, to update our $\theta$ values, and minimize the following loss function:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} exp\left(-y^{(i)} \theta^T \phi(x^{(i)})\right) \quad (4)$$

**Feature Selection**:

We ran forward search on our set of features to simplify our model. We originally had 41 features, and simplified it down to 15. This improved results by reducing over fitting and dimensional of our model.

Forward search is a greedy search over features. You begin with an empty feature set, and try training a model on each individual feature. Then, you choose the best individual

feature, and add it to your feature set. You repeat this process adding each feature to your feature set, and training the model. Each iteration, you add the best preforming feature. This can be done until convergence or until a maximum value is discovered.

**Cross Validation**:

To test each of our classification algorithms, we used leave K out cross validation. This algorithm is used to test the results of a given model on your data set, and returns more accurate results than simply dividing your test set up into a training and test portion of the data.

In this algorithm, you divide your data into K subgroups. You then train your model on K - 1 groups, and use the left out group as your test set, saving the results. You then repeat this process leaving out each subgroup once. Once you have trained leaving out each subgroup, and tested on it, you then average the results, and examine the results.

This method of testing allows for more accurate results because you are able to train on a larger portion of your data each trial, so there is a better chance that your model generalizes to the resulting test set.

## V. Experiments/Results/Discussion

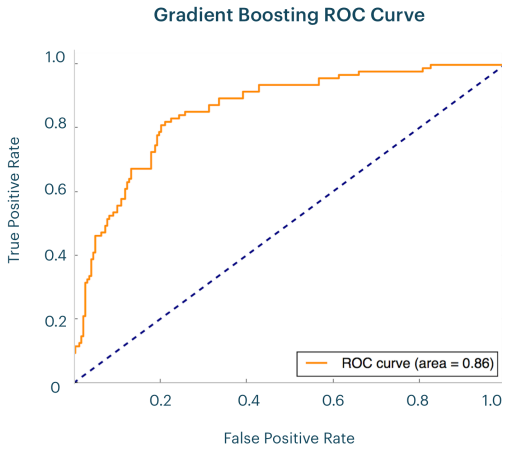



Fig. 2.

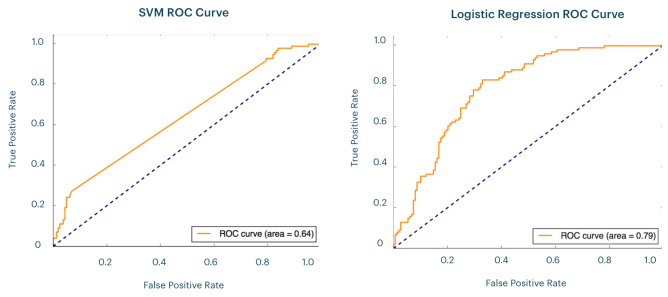



Fig. 3. AUC Values for Gradient Boosting were the highest with 0.86. The SVM AUC was 0.64, and the Logistic Regression AUC was 0.79

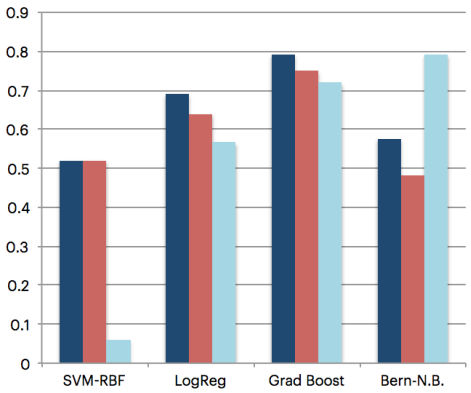



Fig. 4.

**Feature Selection**:

We ran forward search on all 30 keywords, unemployment data, and market sentiment/data. Our top features were:

{ *picket, mob, streets, debate, riot, justice, strike, violence, Bullish 8-Week Mov Avg (market statistic), protest, civil, militarization, movement, demonstration, Bullish* (market statistic) }

We noticed that on larger test sets, our recall fared very poorly. For example, in the test set of 3500 examples, Logistic Regression achieved precision and recall scores of 61.9% and 13.7%, Gradient Boosting achieved 76.5% and 27.3%, and Bernoulli Naive Bayes achieved 13.4% and 43.2%. We then gradually under sampled our data. We maintained the same amount of protest examples, while decreasing the number of non-protest examples. This dramatically improved recall, and gradually improved precision.

Our best results came from Gradient Boosting. Figure 2 shows the ROC curve. With an AUC of 0.86, the Gradient Boosting binary classifier is fairly accurate.

We performed 10-fold cross-validation

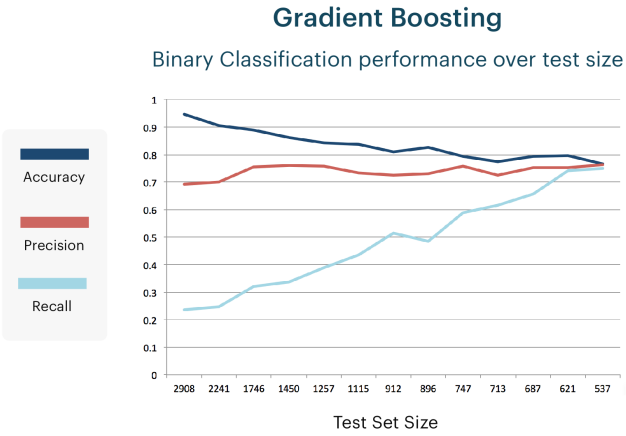



Fig. 5.

## VI. Conclusion/Future Work

In conclusion, we were able to create a reasonable predictor for protest movements in a given metropolitan area. We were able to predict with a pretty high recall and precision after under-sampling our negatively labeled data sets, which means we have created a meaningful predictor that can be leveraged to help all sorts of people.

If we were given the opportunity to continue our work on this project, we would look into social network data. This would potentially give us insights into what individuals actions and sentiment. We would look into Instagram and Facebook data because of our conversations with Twitter. This would allow us to explore a whole different range of features to improve our predictor.