Geoanalysis of the 2016 US Presidential Elections using Twitter Data

Saurabh Sood
University of Colorado Boulder
saurabh.sood@colorado.edu

Abstract

This paper attempts to analyze the 2016 United States Presidential Elections from a geospatial perspective. Analysis is done on a collection of Geotagged tweets. The main purpose of the paper is to find out which agendas of the prospective candidates resonates with voters, which ones may negatively impact a candidates chances. As the collected tweets are geotagged, they are analyzed from a geospatial perspective, and the sentiment for the candidate is plotted on a choropleth map. One of the main observations from the analysis was, that most of the tweets carried negative sentiment, and that people generally rant on twitter, rather than discussing issues related to the election.

I. Introduction

The analysis of the Presidential Elections in the United States is a challenging task, mainly due to the sheer number of factors involved. It is a hard job to statistically predict the outcome of the elections. Sentiment Analysis is one of the techniques which could be used to determine whether a candidate could win the election or not. The candidate's views on various topics, policies, and the sentiment associated with it could go a long way in determine whether a candidate would win or not.

Twitter is a very important social network. What sets it apart from other social networks is the fact that Twitter data is public. It could be viewed by someone who doesn't have a Twitter account. The fact that a Twitter feed could be embedded in other pages is also a significant factor in Twitter being all pervasive in the social media space.

The tweets collected for the purpose of this paper are geotagged, which becomes possible to localize the sentiment in a particular area. For the purpose of this paper, the sentiment on various agendas of the candidates would be gauged based on the origin of the tweet. With this, the popularity of a particular candidate in a specific area/state will be determined. The paper focuses on the main presidential candidates in 2016, namely Hillary Clinton, Bernie Sanders, Donald Trump, and Ted Cruz.

There has been prior work in gauging public sentiment using Twitter. The work of Hao Wang et al[7] deals with large scale Twitter analysis for the 2012 presidential election. This paper attempts to build on that paper, and factor in the geospatial aspect by using geotagged tweets for the analysis.

II. Data

The data for the purpose of the paper is collected from the Streaming API of Twitter. The Streaming API of twitter provides an asynchronous long polling mechanism to retrieve tweets from the Twitter API. The advantage of the long polling is, that it does not hold the connected between the API server and the client. This enables the script to execute code only when tweets are made available. As the US presidential elections carries global interests, a lot of people not from the US could be tweeting about the candidates. To gauge the public sentiment in the US, it makes sense

to restrict the tweets being analyzed to those from the US. For this purpose, the *location* filter was applied in the calls made to the Twitter API, so that only tweets from the USA were retrieved from the API. The *location* filter specifies a bounding box, which will be used to localize the tweets. A tuple of the Southwestern, and Northeast coordinates is needed by the Twitter API. The bounding box used for the purpose of this paper is [(-117.477, 32.496), (-67.2038, 44.6103)].

To retrieve the tweets from the API, the Tweepy[6] library for Python was used. Tweepy provides a consistent access to the Twitter API, and includes OAuth authentication. A Drawback with the Twitter API is that it does not allow for multi-filtering. This means, that if the *location* filter has been applied, then filtering for a particular hashtag, or text does not work. The workaround to this limitation is to filter for particular hashtag's in Python code. Once the geotagged tweets are retrieved, they are filtered for the keywords *cruz*, *clinton*, *trump*, *sanders*, *bernie*. All the tweets pertaining to the presidential candidates are retrieved separately.

Twitter data is very unstructured. People tweet all sorts of stuff, such as links, images, sound clips, videos, and a lot of garbled text. This makes sentiment analysis and topic modelling challenging. In addition to the casual text, a lot of tweets are just retweets. A retweet is a repost of an original post. These also need to be handled properly, before further analysis could be done. For this purpose, before running the analysis, the tweets are preprocessed, so that only relevant text is used. For instance, URL's, images, mentions, usernames, are stripped out of the tweet.

III. Methods

Once the data has been collected and preprocessed, it is ready for analysis. The data consists of 7800 tweets collected for the presidential candidates. As an additional preprocessing step, all the tweets are combined into a single CSV(Comma Separated Values) file, and annotated with a tag, so as to identify that the tweet is for that presidential candidates. This will be required for analyzing sentiment for that presidential candidate.

The system developed can be represented by the following block diagram:

I. Sentiment Analysis

For performing Sentiment Analysis, a Naïve Bayes classifier[9] is trained on a training corpus of 3200 tweets, equally divided among the presidential candidates is prepared. The remaining tweets are training data for the classifier. The training data is manually annotated as being positive or negative.

Naïve Bayes Classifier

The Naïve Bayes classifier is a bag of words classifier, which splits the data based on the word features. It is a conditional probability model. Given a set of k classes, it computes the conditional probability of the given data belonging to all the classes, and returns the class with the highest conditional probability. Formally, this could be expressed by:

$$p(C_k|x_1, x_2...x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

where C_k is one of k classes, x_i represent the set of word features, Z is the scaling factor, $p(C_k)$ is the class prior.

For the purpose of sentiment analysis of tweets, the tweet was divided into a set of words, and the conditional probability of it being in the positive and negative classes was calculated. The class it belongs to has the highest conditional probability. It is highly efficient for text classification.

II. Web Application

As the tweets are tagged by location, they can be geocoded to find out the coordinates required for plotting on a map. For geocoding, the Python *geocoder* module [5] was used. Once the tweets are geocoded, the tweets are written to a JSON file, with the following structure:

```
{
    coordinates: [], //Lat-Lng pair
    candidate: '<candidate>', //Clinton, Cruz, Sanders, Trump
    sentiment: '<sentiment>' // POS or NEG
}
```

A web application was created which reads the JSON[8] file, and is used to plot points, representing the origin of the tweets on a map. The Google Maps API[4] was used to create the required tiles. The tweets are then aggregated by the State of Origin, and the percentage of negative tweets, and positive tweets was calculated using a Spreadsheet. The sentiment for the candidates was visualized using Datamaps[2], which is a D3.js[1] library for creating maps.

IV. Results

I. General Observations

While annotating the tweets for training the sentiment classifier, an interesting observation was made. Most of the tweets carried a negative sentiment. This could be attributed to the fact that there is a tendency to rant on a public social media platform. Supporters of a particular presidential candidate often target supporters of a rival candidate. There are frequent flame wars between supporters of rival candidates. This leads to a significantly large number of tweets carrying a negative sentiment.

On analyzing the tweets, another interesting observation was made with respect to the hashtags used. The hashtags are very consistently used by the supporters or detractors of the candidates. Some of the commonly used hashtags are *#ImWithHer*, *#NeverTrump*, *#FeelTheBern*.

II. Sentiment Analysis Results

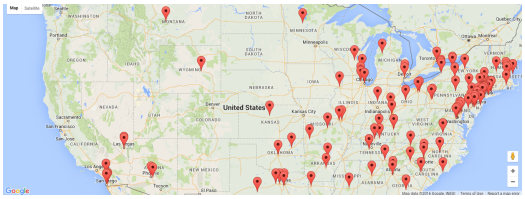
On the test set of 6800 tweets, the Naïve Bayes algorithm gave an accuracy of 68% with 5 folds cross validation. For the purpose of cross validation, 80% of the training data was used for training, and the remaining 20% of the training data was used as test data.
A major observation that was made in the classification was, that most of the tags were classified as negative. This could be explained by the fact that the majority of the training set comprised of negative tweets.

III. Geovisualization

A web application was created in Flask[3]. Flask is a framework for developing web applications quickly. The application exposes the following endpoints:

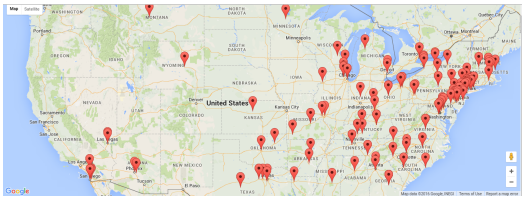
- */*
  This endpoint plots a Google Maps marker on a map, signifying the origin of a tweet. The HTML page provides option to select the candidate, and the sentiment, and the markers change corresponding to the candidate and sentiment selected.
- */<sentiment>/<candidate>*
  This endpoint returns a JSON (JavaScript Object Notation) object for all the tweets of the specified candidate, and the specified sentiment
- */trumpchor*
  This endpoint shows a choropleth map of the the sentiment for Donald Trump
- */clintonchor*
  This endpoint shows a choropleth map of the the sentiment for Hillary Clinton
- */cruzchor*
  This endpoint shows a choropleth map of the the sentiment for Ted Cruz
- */sanderschor*
  This endpoint shows a choropleth map of the the sentiment for Bernie Sanders

The following Google Map showing all the tweets bearing Positive Sentiment for Hillary Clinton:



The following Google Map showing all the tweets bearing Negative Sentiment for Hillary Clinton:

The following Google Map showing all the tweets bearing Positive Sentiment for Bernie Sanders:

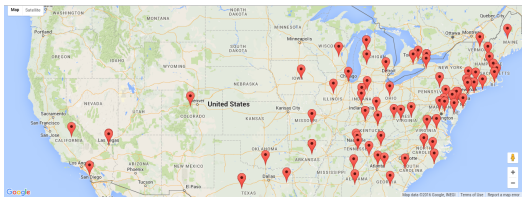


The following Google Map showing all the tweets bearing Negative Sentiment for Bernie Sanders:

The following Google Map showing all the tweets bearing Positive Sentiment for Ted Cruz:

The following Google Map showing all the tweets bearing Negative Sentiment for Ted Cruz:

The following Google Map showing all the tweets bearing Positive Sentiment for Donald Trump:



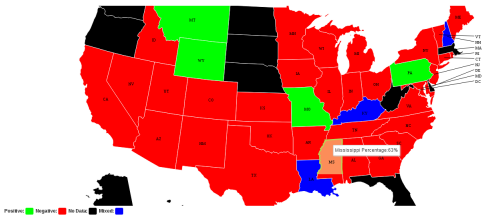
The following Google Map showing all the tweets bearing Negative Sentiment for Donald Trump:

Statewise Sentiment

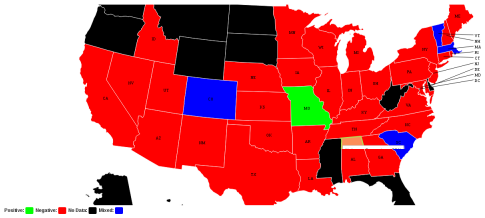
Once the tweets are geotagged, and a sentiment is attached to each tweet, they are grouped by the state of origin. Using a spreadsheet, the percentage of positive, and negative tweets is calculated for each candidate. This information is then used to plot a choropleth map, which shows the sentiment of each candidate in a

particular state.

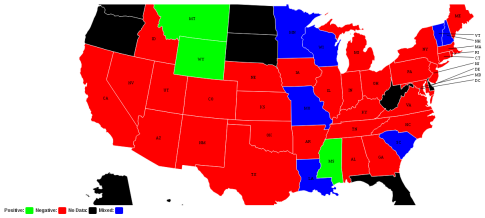
The following Choropleth map shows the statewise sentiment for Hillary Clinton:



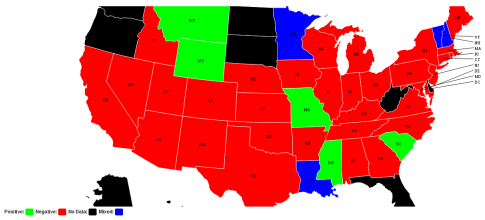
The following Choropleth map shows the statewise sentiment for Bernie Sanders:



The following Choropleth map shows the statewise sentiment for Ted Cruz:



The following Choropleth map shows the statewise sentiment for Donald Trump:



Legend

- Red: Negative
- Green: Positive
- Black: No Data
- Blue: Mixed Sentiment (Equal number of positive tweets and negative tweets)

The choropleth maps throw some interesting observations:

- Most of the states carry a negative sentiment for most of the candidates. This is to be expected, as the number of negative tweets far outweigh the number of positive tweets.
- The sentiment of Bernie Sanders is negative in most of the states. This is a little unexpected, as the number of positive tweets for Sanders is the highest for any candidate. The number of states for which Sanders has a positive sentiment is the least in comparison with all the other candidates.

IV. Sentiment on various topics

While training the data, the sentiment for various topics for the particular candidate was observed and recorded, based on the content of the tweets.

Hillary Clinton

- Negative on lying
- Negative on the crime bill passed during Bill Clinton's presidency
- Negative on the Email leaks
- Negative on fracking
- Negative on the Benghazi attack
- Negative on corruption, and election funding
- Negative on Surrogate Voting fraud
- Negative on the Panama Papers
- Mixed Sentiment on Black Lives Matter

- Positive on Sexism
- Positive on Gun Laws
- Positive on Planned Parenthood

Bernie Sanders

- Negative on supporting the crime bill passed during Bill Clinton's presidency
- Negative on Gun Laws
- Negative on frequent attacks on Clinton

Ted Cruz

- Negative on 14 year citizenship requirement
- Negative on stealing delegates in Colorado

Donald Trump

- Positive over losing Colorado delegates
- Positive on being non establishment
- Negative of not being informed about political process
- Negative on racism
- Negative on Lying
- Negative on Corruption
- Negative on running a negative campaign

V. Conclusions

The analysis of the US Presidential elections resulted in some interesting observations. A claim could be made that Twitter data is unreliable in predicting the outcome of an election, as most of the sentiment is negative. Also, people tend to rant on Twitter, leading to flame wars with supporters of rival candidates. However, Twitter is very effective in gauging the sentiment on various topics.

For the purpose of this paper, the sentiment on various topics was gauged by manually looking at the tweets, which were used for training the data. This could be done automatically by using a Topic Modeling algorithm, and then running sentiment analysis on the result. This is a good candidate for future work.

At this point, the paper doesn't deal with real time Twitter data. In the future, a system could be developed which would gather tweets in real time, perform the classification, recompute the sentiment for the state, and update the choropleth maps.

References

[1] D3.js. D3.js - data-driven documents. `https://d3js.org/`.

[2] DataMaps. Datamaps library for mapping. `http://datamaps.github.io/`.

[3] Flask. Flask microframework for web applications in python. `http://flask.pocoo.org/`.

[4] Google. Google maps javascript api. `https://developers.google.com/maps/documentation/javascript/`.

[5] PyPi. Python geocoding library. `https://pypi.python.org/pypi/geocoder`.

[6] Tweepy. Tweepy library for python. `http://www.tweepy.org/`.

[7] Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pages 115–120, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[8] Wikipedia. Javascript object notation. `https://en.wikipedia.org/wiki/JSON`.

[9] Wikipedia. Naive bayes classifier. `https://en.wikipedia.org/wiki/Naive_Bayes_classifier`.