Political Analysis of Tweets

Sarah Kirby

December 20, 2016

Abstract

The purpose of this paper is to discuss the effectiveness of using a Support Vector Machine to classify the political orientation of post-election tweets. The paper discusses the problem definition, motivation, methods, and results.

1 Introduction

One of the challenges of NLP and sentiment analysis is that word usage and word meanings vary across domains. For example, if a person says they are scared, this might indicate a positive sentiment if they are watching a horror movie. However, it would indicate a negative sentiment if they are talking about an election.

The political domain is especially complicated, because the language used by politicians and their supporters changes every election. In the 2008 election, democrats were talking about "change" and joking about Sarah Palin's soundbytes. In the 2016 election, Hillary Clinton made the term "deplorables" popular. The phrase "nasty woman" also became popular after Donald Trump used it to describe Hillary Clinton. However, Clinton supporters co-opted the phrase and turned in into a positive slogan. A simple sentiment analysis would have tagged the word "nasty" as negative ignoring the context. This demonstrates the clear challenges with performing sentiment analysis for political purposes.

This paper will address this issue by presenting an alternative to sentiment analysis for politics. Instead of analyzing text for sentiment about a candidate or subject, I will explore the possibility of analyzing the language trends. I will look at whether democrats use certain words or phrases more than republicans, regardless of their sentiment. To accomplish this, I will use a Support Vector Machine (SVM) to classify tweets.

1.1 Problem Definition

The problem I want to address is whether we can predict a person's political orientation from their language. To simplify this, I have chosen to analyze tweets to predict whether a person is pro-Trump or anti-Trump. I selected tweets with the hash-tags #MAGA, and #NotMyPresident. MAGA stands for Make America Great Again, which was Trump's slogan during his campaign. Tweets with #MAGA will be assumed to be pro-Trump. The #NotMyPresident hash-tag began as a protest of Trump's election, so these tweets are assumed to be anti-Trump. I will look at the effectiveness of using an SVM to classify these tweets.

1.2 Motivation

The motivation for this project began with the 2016 U.S. Presidential election, and the campaigns leading up to it. Throughout the campaign period, Hillary Clinton was ahead of Donald Trump in the polls. Towards the end of the race her lead dropped, but she was still the predicted winner. When Trump won the election it shocked many people, even experts and pollsters. It appears that we need better methods of keeping track of the political distributions in our country. This motivated me to look into analyzing social media for political purposes.

2 Related Work

The main inspiration for this project was the study, "Analyzing Twitter Sentiment of the 2016 Presidential Candidates," done by researchers at Stanford University[2]. In this study, tweets were collected and analyzed to predict their sentiment about several candidates. The candidates analyzed for were Bernie Sanders, Hillary Clinton, Donald Trump and Ben Carson. Only tweets with emojis were collected for this study. The emojis were used to assign initial sentiment labels for training. The "bag of words" method was used for feature extraction, and words that didn't indicate sentiment were removed. The study compared the results of several classification algorithms. The results showed that the SVM outperformed Naive Bayes and Nearest Neighbors classifications. The highest accuracy for the SVM was 53%.

In my study I will analyze tweets for the combinations of words used, rather than the sentiment of those words. I will use the "bag of words" method for feature extraction, but I will not remove words that don't indicate sentiment. I will collect and label tweets based on hash-tags rather then emojis, and I will remove emojis from all tweets collected to analyze text only. I will also use an SVM, but I will perform binary classification instead of multi-class. Finally, I will examine the effects of feature reduction on my SVM accuracy.

3 Methods

3.1 Tweet Collection

I used Twitter's Search API to collect tweets. Twitter limits search queries to tweets from the last 7 days, so I could only collect tweets from after the election. I limited my search to January 13$^{\text{th}}$ and 14$^{\text{th}}$ of 2016, and I filtered out re-tweets and replies. In total collected 5,000 tweets containing #MAGA and 5,000 containing #NotMyPresident. I used MongoDB to store and clean the tweets. To clean the tweets I removed emojis, urls, usernames, punctuation and any Unicode characters that would not translate to ASCII. I converted all the text to lowercase.

3.2 Feature Extraction

I used the "bag of words" method of feature extraction for NLP. This method ignores word order, and treats each occurrence of a word as a feature [1]. To implement this method, I iterated through the 4,000 training tweets and stored all unique words in a CSV file. I then used this file to create a sparse array for each tweet, where the occurrence of a word is marked by a 1 in the array.

To build the full feature list I analyzed the clean versions of tweets. I excluded words that were shorter than 3 characters. The total number of words used in the training tweets was over 7,000. To reduce this, I eliminated words that were only used once. The total number of features was then 3,382.

3.3 Classification

I performed binary classification using MATLAB's fitclinear function. By default, the fitclinear function uses an SVM. I used fitclinear instead of fitcsvm to improve performance, since my data was very sparse. I trained with 2,000 examples of each class for a total of 4,000. I randomized the training data as well. I tested the SVM with 3,000 of each class for a total of 6,000 test examples.

3.4 Dimensionality Reduction

To further explore the use of an SVM for classifying political text, I looked at the effects of feature reduction on the accuracy of the SVM. To do this I compared two methods of dimensionality reduction. First, I simply ordered the features by the number of occurrences in the training data. I then trained and tested the SVM on the most-used features, incrementally increasing the number of features from 20 to 3,382. Second, I used MATLAB's svd function to mathematically reduce the dimensions. I compare the results of these two methods in the following section.

4 Results

4.1 Baseline

My classifier has a baseline accuracy of 74%. This is using all the features extracted, and the default settings for the SVM. To get a better idea of how the SVM is classifying tweets, I collected the top five tweets for each hash-tag. The top tweets are the tweet that had the highest reported probability of being in that class. Below are the cleaned versions of the tweets.

Top #MAGA Tweets

> welcome to the 1st annual hunger games
>
> false flag rhetoric compels insanity from dems no hacking in election just butthurt libretards spewing hate aga
>
> better run to your safespace thanksjackasses trumptrain 2a
>
> war veteran bored to death by retirement starts first day at new job presidentelecttrump
>
> i think anyone running for prez and wh jobs should get a mental exam america got a bunch of nuts about to run this country

Top #NotMyPresident Tweets

> 19 electors join call for russian hacking briefing before voting
>
> this is serious kanyewest um no
>
> funny way of warning us bout what the peoples government has access to your privacy in january 17 watch your ba
>
> thanx for followback
>
> because after all this is a dictatorship electors better not cross trump resist trump russia

These tweets highlight some of the challenges in classifying tweets. The last tweet in the #MAGA list was actually a #NotMYPresident tweet that was misclassified. Several of these tweets, although classified to the right hash-tag, don't show the political orientation you would expect. Some of them don't show a political orientation at all. This is likely because twitter users often use hash-tags to have their tweets seen on searches for the hash-tag, or to get attention from users of the hash-tag. Using a hash-tag does not necessarily mean that a tweet supports the hash-tag. This is just one of the many challenges of using twitter data, particularly in using hash-tags for classification.

4.2 Dimensionality Reduction

Figure 1 shows the effect of the first reduction method on the accuracy of the SVM. Figure 2 shows the effect of the mathematical reduction with svd on accuracy. When comparing the graphs, you can see that both achieve similar accuracies. The data reduced by word usage drops off below 500 features. The accuracy of the SVD-reduced data retains accuracy at lower dimensions, dropping off below 100 features.

Impressively, the accuracy for both sets of data always remains above 60%, even with as low as 20 features. I think the reason for this may be that in my feature extraction, I included hash-tags other than the ones used for classification (#MAGA and #NotMyPresident). Other hash-tags that appear in the most-used features include #PresidentElectTrump, #TCOT (Top Conservatives On Twitter), and #TrumpTransition. It is likely that these other hash-tags serve as very polarizing features.

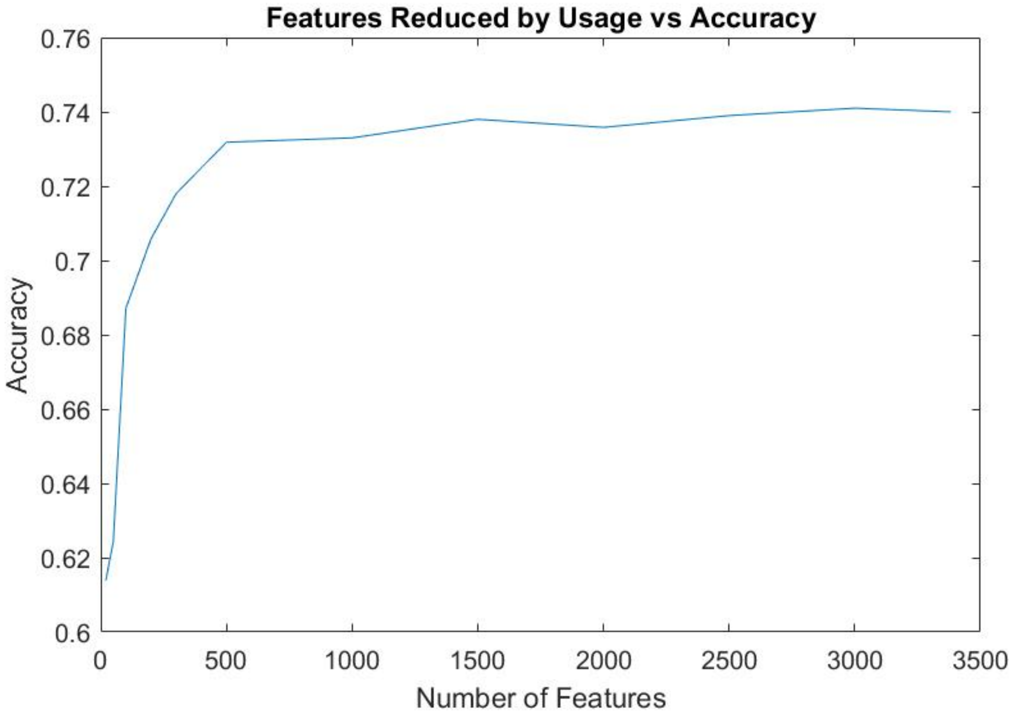

Figure 1: Limiting features by usage

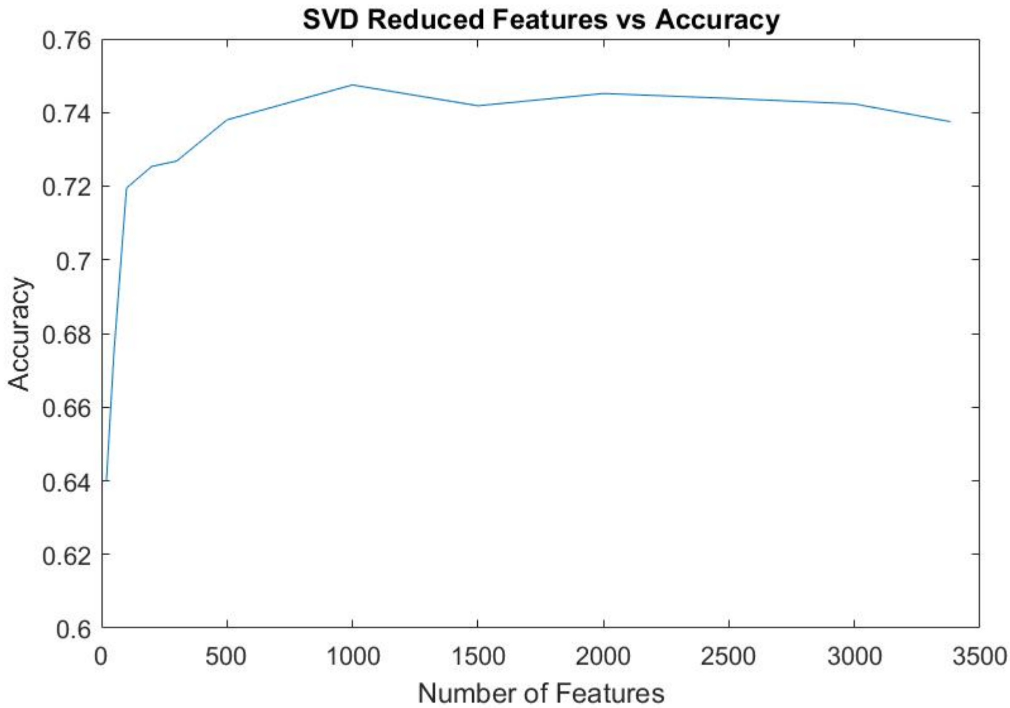

Figure 2: Feature reduction by SVD

5 Conclusion

Overall, I was very impressed with the accuracy of this classification, and with the limited effects of feature reduction on the data. For a future study I would be interested in seeing how the classification performs on data with all hash-tags removed. I would also be interested in seeing how well this method can classify general political orientation, as opposed to just the political stance on Trump's election.

References

[1] Bag-of-words model. Wikipedia, 2016.

[2] Delenn Chin, Anna Zappone, and Jessica Zhao. Analyzing twitter sentiment of the 2016 presidential candidates. Technical report, Stanford, 2016.