# Ideology in 140 Characters: A Machine Learning Predictor of Political Leanings in Twitter Users

Anton Apostolatos, Leonard Bronner, Cristian Cibils.

antonaf, lbronner, ccibils

November 14, 2014

CS221: Artificial Intelligence - Stanford University

**Abstract**

Predicting people's political views has always been a critical issue for campaign managers. Our project focuses on predicting the ideological characteristics of Twitter users on a negative zero point five to positive point five scale, where -0.5 represents liberal and 0.5 represents conservative. We achieve our goal by leveraging the network characteristics of Twitter and feeding them into a Machine Learning model that learns from a set of features (followers, following, etc.) in the accounts of representatives where the output, or $y$, is their political ideology score as given by GovTrack.us.

## 1 Problem Background and Implications

Initially we decided to tackle the problem of predicting Twitter user's political views as an unsupervised problem. Our first idea was to identify how users stood on different social, political and economic dimensions using their Twitter information, placing them in a multidimensional space. This would allow us to figure out the optimal positions of "politicians" by minimizing the distance to the maximum amount of users and at the same time, project the users onto a one dimension Democrat-Republican scale. However, we quickly found out that it was impossible to get enough data in order to truly classify their views on different dimensions, and when we did get information, it was dimensionally too sparse to be of use, so we abandoned that approach.

Our next idea was to cluster users using only tweet data. We used a uni-gram model (which we planned to later expand to try bi- and tri-gram models) to make word feature vectors, ignoring following and follower data for the time being. We settled on using cosine similarity for distance and then used a simple K-Means algorithm to figure out the "optimal position" for politicians by minimizing the distance to users.

However, after implementing a successful test run we realized there was little room for advancement given the current model, so we decided to re-examine our initial goal. In our research we came across the paper Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data by Pablo Barberá, which inspired us to turn this into a supervised problem.

For this to be possible, though, we needed an annotated dataset. Thus, we began by linking representative's Twitter accounts with their respective information on the website GovTrack.us, which mapped each representative to a specific point on an political ideology spectrum. Our intuition was that if we can map features to these scores, we could extend our model to regular Twitter users and predict their place on said spectrum, in turn predicting their political leaning.

The implications for this project are clear. In a time where directed advertising and marketing are extremely important and sophisticated, predicting political affiliation is a powerful tool. Furthermore, the results demonstrate how easy it is to predict individuals political opinions from public data, a relevant finding in a time where data privacy has become an important issue.

## 2 Data Collection

### 2.1 Collection of tweets

In order to collect Twitter users' data we developed an interface with the Twitter API. Given a list of user handles or Twitter-specified user identification numbers we were able to collect all of the necessary data for each Twitter user, including information on who they follow, information on who follows them, as well as a comprehensive list of all of the user's tweets and retweets.

### 2.2 Politician collection and analysis

We gathered the list of the Members of Congress's Twitter handles and political ideology using information from GovTrack. Each politician's leaning is categorized within a defined ideological spectrum with domain [0, 1] (which we then altered to [-0.5, 0.5] before running the learner). The closer someone's ideology is to 0 the more democratically-leaning a person is. Inversely, the closer someone's ideology is to 1 the more republican-leaning a person is. We analyzed all politicians in the 113th United States Congress, whose ideology is mapped out in Figure 1.

GovTrack computes this data using dimensionality reduction, examining each Member of Congress's pattern of sponsorship and cosponsorship of bills. The process itself doesn't take into account the bill's contents or the party affiliation of members, but rather is able to infer underlying behavioral patterns which, in general tend to correspond with the concept of right-left ideology.

By parsing this dataset, along with a list of all Members of Congress' personal information, which included their full name and their Twitter handle, we were able to create a map of each Member of Congress' Twitter handle to their political ideology. In order to capture each congressperson's Twitter information we ran the list of all Members of Congress through the Twitter user collection system described in the previous section.

## 3 Process

The first step in our process was deciding what would serve as an appropriate baseline. We eventually set our baseline to be the correct classification of Twitter accounts as if the problem were not a regression problem, but rather a classification problem (namely, correctly guessing which side of the spectrum a given user stood at). On the other hand, our oracle became exact prediction of people's political views on the ideology spectrum (or rather, a 0.0% average error on our testing set). In terms of performance, we hoped to achieve a O(NM) performance for our Baseline where N is the number of training examples and M is the number of features for that example. Furthermore, for the oracle, we hoped to achieve O(N), where features are extracted in constant time.

The next step of the process was deciding which features to extract. Initially, we decided we would leverage the network properties of the Twitter graph and use politician's followers and following information. The intuition behind this was that people of similar ideological standpoints not only followed similar people on Twitter, but were also followed by similar people.

We used the Twitter IDs of the accounts followed/following our training examples as feature templates. We tested this feature extractor and were very pleased with the results (more on the results section). Enthused by our results, we decided to add tweet information as well, in order to leverage the semantics behind the politicians' discourse. We approached the challenge by implementing simple word counts for each politician, average word length, and number of hashtags used. However, we were disappointed to find that the data was too noisy. What we found by this, was that even though politicians may stand of different sides of an issue, they use the same vocabulary. Unwilling to give up, we found that a bi-gram model for tweet parsing improved our performance. We elaborate on these results in the section below.

The final step in our process was selecting the optimal algorithm. In order to accomplish said goal, our approach was to use gradient descent on our newly extracted twitter features to arrive at the solution to our linear regression problem.

# 4 Results

Results In order to get information and insight on the different models and changes made to the feature extractor and hyperparameters, we implemented a system which ran repeated random subsampling validation with ten iterations. In essence, in each iteration the program would randomly select 80% of the set of politicians and use it as training data, while the remaining 20% would be used as the testing data. The system would then calculate both the average baseline and average absolute errors. The average baseline error would be the percentage of instances in the testing data where the classification placed the user in the incorrect side of the spectrum (namely, when a user real ideology was negative and the predicted ideology was positive, and vice-versa). The average absolute error is the average distance in the political leanings spectrum between a user's predicted ideology and their actual ideology.

## 4.1 Feature Extractor Testing

Initially our feature extractor only captured following data, which means we created a feature vector where each user that the politician was following was a dimensions. This means the implicit feature template had every user on Twitter in it, where a one meant the user was being followed by a politician and zero meant otherwise. This gave us textbfan average error of 0.157 and a **baseline error of 0.103**.

We then tried to do followers only. The concept of followers is the same as following, except that it only took into account the people that were following the given users. This took a lot longer than following, as politicians tended to have a lot more followers than people that they follow. **The average error here was 0.162** and the **baseline error was 0.207** - substantially worse than with only following information.

As a next logical step, we decided to combine both following and followers. While this made the feature vectors very long, and thus took quite long to run, we noticed that the results here were the best. It gave us an **average error of 0.135 and a baseline error of 0.076** (this is extremely good - we can pretty much predict with great accuracy if someone is left-wing or right-wing). It is also the most useful, because in order to use this on an average Twitter user, the model needs to have seen as many users and possible. This decreases the possibility of not being able to give someone an ideological score due to no intersection between his followers and following the database of user weights we have.

Figure 2 in the appendix presents a graphical representation of this information.

### 4.2 Tweet Parsing

One of the final modifications to our feature extractor was the inclusion of a bi-gram model for tweets and retweets. We decided that the likelihood of pairs of words appearing together would be significantly more representative of the politician's point of view than just the occurrence of a single word. Moreover, we thought about adding a tri-gram model, but the memory punishment was too steep for our systems (although it leaves an interesting segway for further work). Our final results when using tweet parsing were 0.1309 for average error and 0.229 for baseline error. We believe that this increase in the baseline error is because the large number of tweets causes feature templates to be orders of magnitude larger than the number of followers/following, thus causing them to be of less significance when updating the weights. We were pleasantly surprised to find this model led to very slightly better average error, but were disappointed with the cost/benefit relationship between this increase in accuracy and performance time.

### 4.3 Hyperparametric Testing

We then moved on to Hyperparamater testing. In order to do this, we took around 10% sized validation set from former Members of Congress, whose data we found on GovTrack.us. We ran multiple iterations with followers and following, as this had been our best feature combination. We changed the number of iterations and the eta, and came to the conclusion that a number of iterations of 20 and an eta of 0.00011 gave us the best result.

Please look at figures 3 and 4 in the appendix for a graphical representation of the trials we went through.

## 5 Conclusion

Our results indicate that the network properties of the Twitter graph offer a great amount of insight into predicting people's political views. Using the data that is publicly available we can, with a great degree of certainty, binarily predict the political leaning (Democratic vs. Republican) and we can even tell "how" much of a Democrat or Republican a user is - even if this is not fully refined.

Our most surprising finding, however, was how little our tweet extraction helped. For all practical purposes, in terms of computation power spent into the predictions, we realized that it makes significantly more sense to not compute the bi-gram analysis, given that it only marginally increases our optimal accuracy.

However, our greatest accomplishment in this project is being able to correctly classify people's political leanings with very close to perfect accuracy (solidly satisfying our benchmark for the baseline), making our algorithm versatile as both a classifier and a linear predictor.

## 6 Future Work

There is clearly a lot of opportunity for future work here. Firstly, it would be extremely exciting to incorporate tweets in a more meaningful way. At the moment it seems that it is neither possible, nor helpful. However, maybe there are ways to decrease the error further. One possibility would be to only use words that we know are politicized (e.g. "Obamacare", "Immigration" or "Gun Control"). This would eliminate a lot of the noise that we are currently seeing in our data. Another step would be incorporating more social media channels into our project. While Twitter is very widely used, we think that access to people "fan pages" and "groups" on facebook would allow for a more nuanced analysis of someone ideology score. Finally, we would like to move this project back towards an unsupervised problem. In Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data, Barberá approaches the problem by choosing 500 organisations, and clustering them in between users, iterating back and forth between the users and organization to find the optimal position. This may be a possible approach for us, clustering the politicians that we have and the users we want to classify. This would give us more generality when it comes to expanding on this project outside of the US and making it general enough for future congresses. As it stands at the moment, relearning every few years is necessary. However, if we managed to find an unsupervised approach, classifying people politically could become a generally solved problem.

# 7 Citation

GovTrack.us. 2013. Ideology Analysis of Members of Congress. Accessed at https://www.govtrack.us/about/analysis.

Tauberer, Joshua. 2012. Observing the Unobservables in the U.S. Congress, presented at Law Via the Internet 2012, Cornell Law School, October 2012.

# Appendix

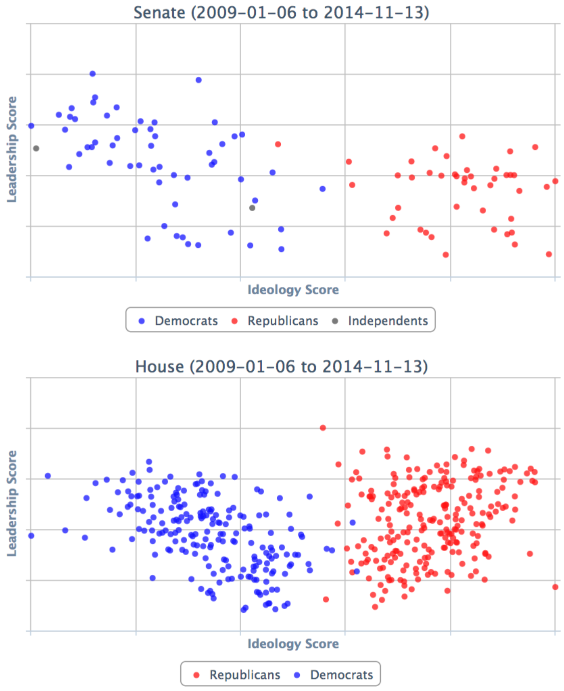

Figure 1: The scatter plot maps each congressperson in the 113th Member of Congress to their political leaning and their leadership.

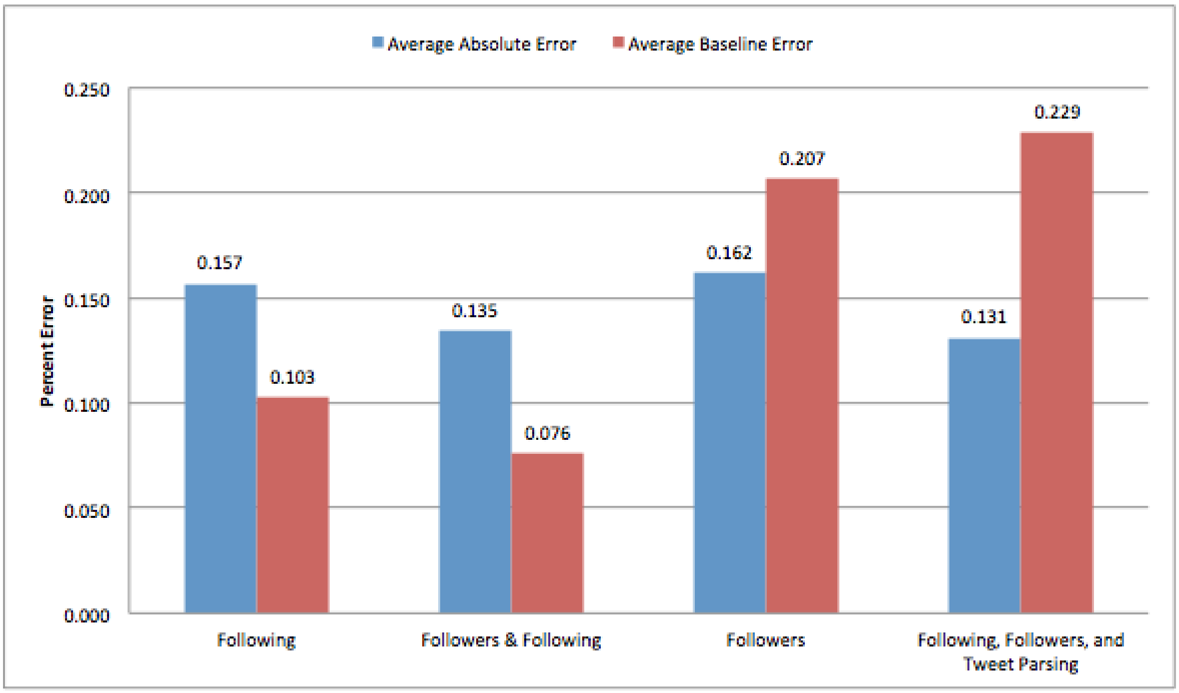

Figure 2: The average baseline and absolute errors for each model of the feature extractor.

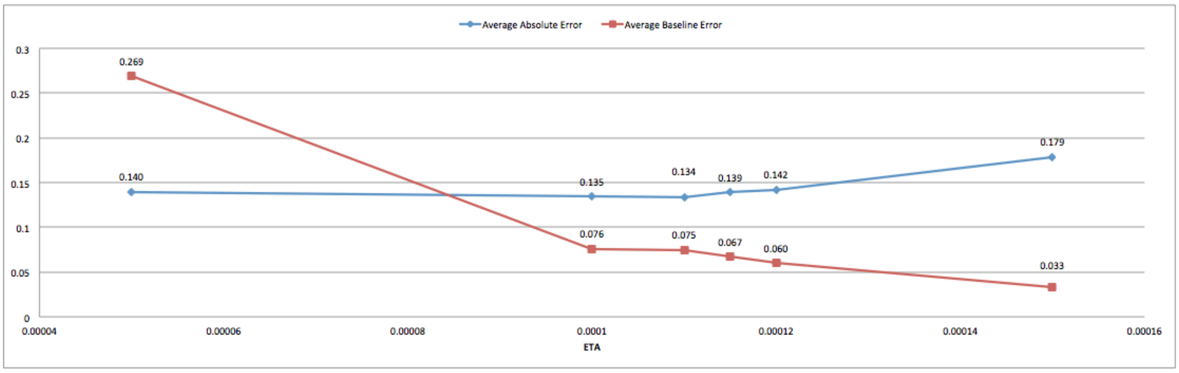

Figure 3: The average baseline and absolute errors for various values of the ETA

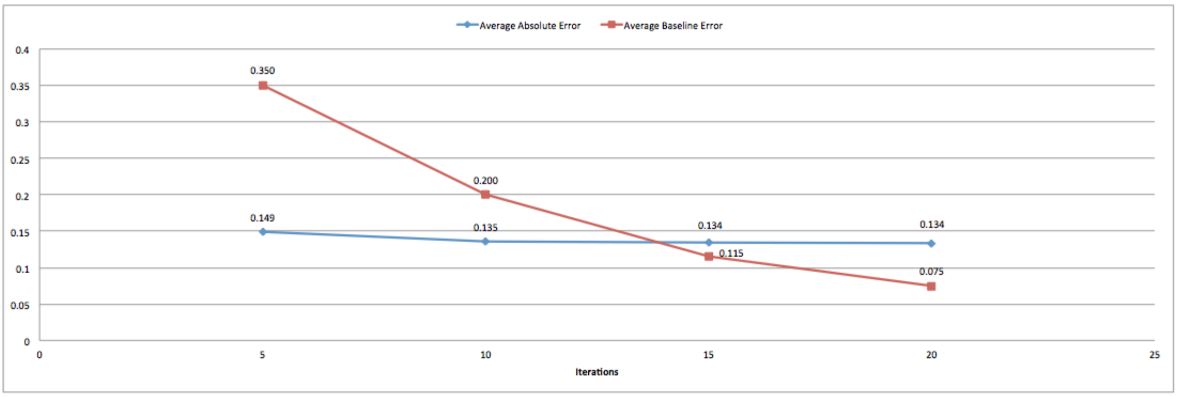

Figure 4: The average baseline and absolute errors for different number of iterations.