Using Statistical Methods to Classify Political Speech

Zoe Koch

5 December 2017

Crowdflower's Data For Everyone Library, has provided text of 5000 messages from politicians' social media accounts, along with human judgments about each post's audience (national or the tweeters constituency), bias (neutral/bipartisan, or biased/partisan), and finally tagged as the actual substance of the message itself (options ranged from informational, announcement of a media appearance, an attack on another candidate, etc.). This project uses statistical learning to classify political speech in the form of social media messages based on their text content. A variety of methods were employed to build a predictive model, including decision trees, naive Bayes classification, and logistic regression, among others. The majority of methods had a variety of issues with the data being input. In the case of naive Bayes it failed to calculate non-zero probabilities and was as a result simply grouping randomly. This same issue occurred with support vector machines.

Initial success, at least in visualizing what words were indicative of certain classes, was seen using simple decision trees. You can see that the presence or absence of the word 'obamacare' was highly indicative of whether a message was partisan or not (Figure 1). Additionally it's clear that 'live' and 'attend' are indicative of a message being meant for an audience of the politicians constituency, whereas the words 'prevent', 'night', and 'world' are indicative of a message being meant for a national audience (Figure 2).

Unfortunately these methods barely had classification rates higher than that of the no information rate. In the case of classifying on bias, the No Information Rate was 0.7382 and the tree correct classification rate was 0.747. The confusion matrix (Table 1) indicates that it classified nearly all of the messages as being neutral, and picked out those which contained the word obamacare to classify those as partisan. Classifying by audience had even worse results (Table 2) with a classification rate slightly below the no information rate. Again, from the confusion matrix it seems that simply messages with the words 'live' and 'attend' were classified as being for constituency with the rest being classified as nationally targeted messages.

A random forest approach was attempted, and little improvement was seen. In the case of classifying by bias, a correct classification rate of 0.755 was achieved. This is slightly above the previous 0.747 rate of the single tree, but it is not a significant improvement. The confusion matrix for both this and audience show that they are similar to those of a single tree (Table 3 and Table 4). For classifying by audience the random forest ended up simply classifying all of the messages as being for a national audience. This is the no information rate, but that resulted in an improvement over a single tree.

Logistic regression failed to converge, but a form of penalized logistic regression called Firth Logistic Regression showed promise. The Firth Logistic Regression is the equivalent of using a Bayesian Jeffrey's Prior on parameters for the regression, and always results in finite and non-zero results through the addition of a fixed value to the parameters that reduces bias from the MLE estimators of the regression coefficients. This method was able to correctly classify into both bias and audience with rates of 0.922 and 0.972 respectively (Table 5 and Table 6). The ROC

curves (Figure 3 and Figure 4) show very good True Positive Rates vs False Positive Rates for both Audience and Bias. This means that these models can be used to classify political messages fairly accurately and can likely be applied to political tweets not included in the data set.

In this particular case it's difficult to explain *why* these models are good at classifying, and incredibly difficult to determine what's going on in the model. 38 different variables are taken into account in order to perform the classification (the absence or presence of particular words) and many of them are non-significant in the model, but their removal results in reductions in model accuracy. It's clear that words like 'obamacare', 'job', 'senate', 'vote', and 'bill' are all important when trying to determine whether a particular message is biased or neutral, but saying something like "The presence of the word 'job' in a message, holding all other words constant, increases the log-likelyhood of that message being biased by 0.4795" doesn't exactly elucidate the nature of bias in these messages. While you could make some conclusions, saying that words we associate with political agendas may be more likely to be associated with bias, the model is a little more complex than that. When it comes to predicting the audience the same seems to hold true, but other words become important in the decision. Again, these models are better suited for prediction than interpretation, and shedding light on what makes political speech targeted towards a particular group or partisan if difficult to do.

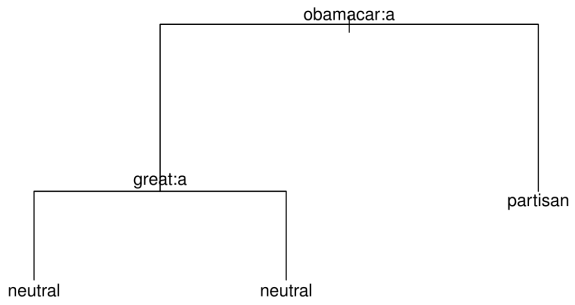

Figure 1: Decision Tree for Classifying Bias

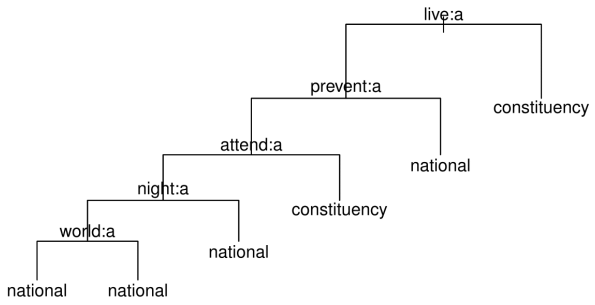

Figure 2: Decision Tree for Classifying Audience

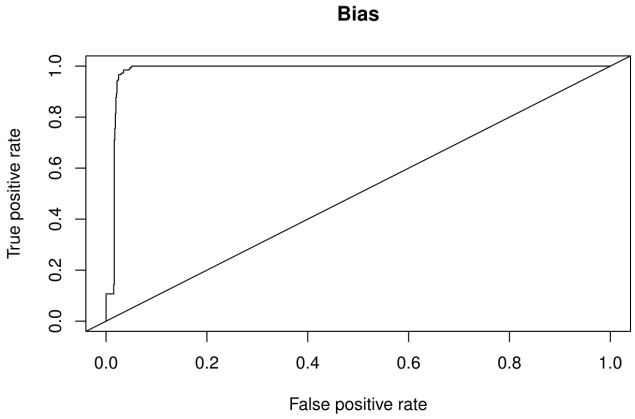

Figure 3: Firth Logistic Regression, Bias

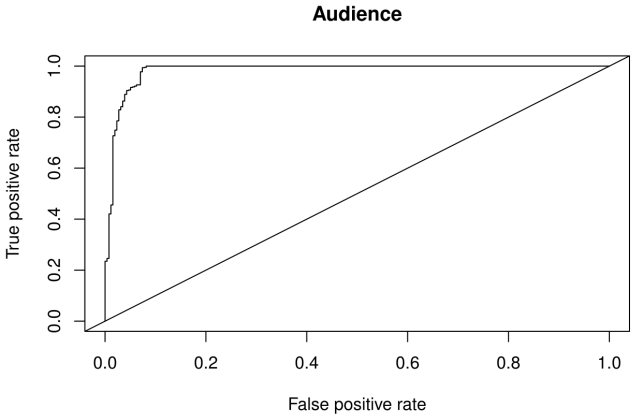

Figure 4: Firth Logistic Regression, Audience

	Neutral	Partisan
Neutral	914	308
Partisan	8	19

Table 1: CM for Tree on Bias

	Constituency	National
Constituency	29	82
National	228	910

Table 2: CM for Tree on Audience

	Neutral	Partisan
Neutral	908	292
Partisan	14	35

Table 3: CM for Random Forest on Bias

	Constituency	National
Constituency	0	0
National	257	992

Table 4: CM for Random Forest on Audience

	Neutral	Partisan
Neutral	905	80
Partisan	17	247

Table 5: CM for FLR on Bias

	Constituency	National
Constituency	238	16
National	19	976

Table 6: CM for FLR on Audience