Predicting NRA Ratings from Congressional Speeches

Anqi Ji anqi
Allison Koenecke koenecke
Julia Olivieri jolivier

Abstract

We implemented machine learning models to predict the rating the NRA would give a congressperson based on one of their speeches in Congress. Using data from 2011 through 2016, we used a frequency-based featurization for the speeches to perform this 4-category classification task. Out of three Naive Bayes methods and an SVM, the SVM performed the best, with 60.2% accuracy. We compare this to a multinomial logistic regression without using speech data based on party, state, age, and gender, which achieves 96.3% accuracy.

1 Introduction

With the tragic upswing in gun deaths in recent months (including the deadliest mass shooting in modern US history this October), gun control has again been pushed to the forefront of American politics. In this project we use congressional speeches to predict congress peoples' stances on gun control as measured by the letter grades assigned to them by the NRA (National Rifle Association). This will allow us to fine-tune gun stance predictions past the standard assumption, which is a basic 2-sided classification by party (Republican or Democrat). As such, we will be able to determine whether text-based congressional statements are helpful or irrelevant in predicting gun stances.

For this project, our inputs use two data sources: personal attributes of the congressperson as of 2013 from ProPublica [5], and Congressional records (for both the House and the Senate) for every day of 2011 through 2016 [1]. These data were cleaned, featurized to multinomial matrix format, and joined by congressperson. We then developed classifiers that take in a political speech as input and return as output a prediction of whether the congressperson who gave that speech would be rated an A, B, C/D, or F by the NRA. The methods used to classify speech and congresperson data include: three variations of Naive Bayes, a Support Vector Machine, and multinomial logistic regressions.

2 Related Work

The two major branches of research in text classification are featurization and classification algorithms [6]. A good featurization not only makes large problems computationally efficient but also improves accuracy significantly. Two commonly used featurization methods are Best Individual Features (BIF) [2] and Sequential Forward Selection (SFS) [9] [10]. BIF is much simpler and faster to implement, but SFS usually gives a better accuracy.

For classification algorithms, Naive Bayes assumes all the features of an example are independent of each other. This assumption has been shown to perform surprisingly well for text-classification tasks. Naive Bayes can be implemented using Multivariate-Bernoulli or Multinomial methods, where the latter performs better for small vocabulary sizes [8]. Continuous variables can be classified using Gaussian Naive Bayes [7]. If avoiding the Naive Bayes assumption, Support Vector Machines can be used and have been show to often achieve higher accuracy [11].

Logistic regression, Naive Bayes, and Support Vector Machines have previously been used to classify political campaign speeches into different parties, regions and election year [4]. Previous results yielded accuracy in the 90% range.

Table 1: Dataset Sizes

Dataset	Size (Speeches Only)	Size (NRA Data Only)	Size (Combined Data)
Training set (80%)	17,890	252	17,428
Test set (20%)	4,473	64	4,357
Training set (2011-2015)	18,990	–	18,412
Test set (2016)	3,373	–	3,373
Total	22,363	536	21,785

3 Dataset and Features

We downloaded the Congressional Record for every day Congress was in session from 2011 through 2016 [1]. We then collected every speech from each Senator or House Representative and stemmed them using the `nltk` python package after removing all non-alphabet characters and converting to lower-case. Words are included as features if occurring with varying frequency ranges over all speeches in the course of the six-year period. The number of words to use as features is the length of the vector created for each speech.

Further, we downloaded data records [5] for each congressperson from 2013 that include their NRA letter-grade ratings (A through F), as well as personal features such as gender, birthdate, political party, state, etc. For each data row, we generated a classified version of the NRA rating (originally with 15 different categories), wherein we made four-element (A, B, C/D, and F) and three-element (A/B, C/D, and F) classifications to be used as the outputs we attempt to predict. Further, categorical variables (such as gender, state, and party) were made into indicator variables so that they could be used as features in regression analyses.

To implement hold-out cross-validation, we first used 5-fold validation on a random 80% of the dataset, and tested on the remaining 20% of the data held out. Secondly, we used 5-fold validation on all speech data from 2011 through 2015, and tested on the remaining data held out from 2016. The data counts of training and test sets are described in Table 1.

Before exploring machine learning methods, we performed several top-line checks of our data. We confirmed that the distributions of NRA grades and letters are as expected (relatively bimodal) based on media coverage, and also highly polarized based on political party (see Figure 1). Further, we found that "gun" was within the top 600 frequently-mentioned words throughout the congressional speeches (out of 48,798 words occurring overall), occurring 3,478 times over the six-year time period.

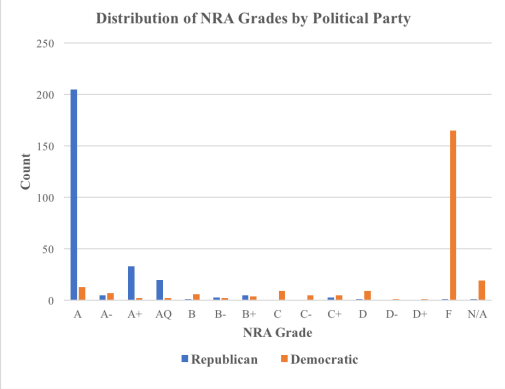

Figure 1: NRA Grade Distribution for Republican and Democrat Congresspeople in 2013 [5]

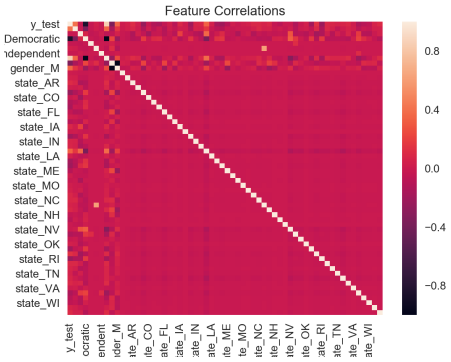

Figure 2: Correlation Between Features Applied to Regression Models

4 Methods

In this paper, we implemented three different Naive Bayes models, a support vector machine and multinomial logistic regression to predict a congressperson's opinion on guns. We used the Naive Bayes models and support vector machines on congressional speeches and the logistic regression on congressperson attributes (political party, state and etc.) We define x_i as the i^{th} feature of a document,

C_k as the k^{th} class, x_l as the l^{th} document, m as the total number of documents, N as the total number of features, and K as the number of classes.

All Naive Bayes models are based on the assumption that all features are independent with each other [3][8], i.e.

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

This assumption suggests that the correlation between each word is very small and we ignore joint information from longer phrases. This yields a simpler algorithm, especially when feature vector is in very high dimension [8], wherein the classification is the optimization problem $\hat{y} = argmax_{k \in 1,...,K} p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$. The probabilities p depend on the type of Naive Bayes model chosen, and involve applying Bayes' rule. We implement the following three Naive Bayes models.

Both Multivariate Bernoulli and Multinomial models take discrete values as input. For Multivariate Bernoulli models, a text is featured by a vector of binary values indicating if a word occurs or not in the document. The probability of a feature i being in a class is $p(x_i|C_k) = P(i|C_k)x_i + (1 - P(i|C_k))(1 - x_i)$

On the other hand, Multinomial Naive Bayes calculates this probability by the number of occurrences of word i in this document, which can be expressed as $P(x_i|C_k) = \frac{N_{C_k i}+1}{N_{C_k}+n}$, wherein we have applied Laplace smoothing to avoid the case that a word has not yet appeared in the document, $N_{C_k i}$ is the number of times the i^{th} feature appears in class C_k, and $N_{C_k i}$ is the total count of all features for class C_k.

Lastly, the Gaussian Naive Bayes model is associated with continuous input variables and assumes that the distribution of the features follows a Gaussian distribution, i.e. $p(x_i|C_k) = Gaussian(\mu_{x_i,C_k}, \sum_{x_i,C_k})$, where $\mu_{x_i,C_k} = \frac{1}{m} \sum_{C_k=l}^{y_l} x_i^l$ and $\sum_{x_i,C_k} = \frac{1}{m} \sum_{C_k=l}^{y_l} (x_i - \mu)^2$.

Aside from Naive Bayes models, Support Vector Machines are designed for high dimensional feature vectors. In SVM, a hyperplane is optimized to maximize its distance with the nearest data point in all different classes. There are various choices in Kernel functions. In this paper, we use the `sklearn` default kernel rbf, which can be mathematically expressed as $k(x, x') = exp(-\gamma||x - x'||^2)$ for $\gamma > 0$, and the classifier is $f(x) = \sum_i^N \alpha_i y_i k(x_i, x) + b$. Generally, the primal formulation for the SVM optimization problem is $\min \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i$ s.t. $\xi_i = \max(0, 1 - y_i f(x_i))$ and $y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i$ for $i = 1, ..., N$.

Lastly, Multinomial Logistic Regression is used to predict classifications based on a politician's attributes: political party, state, age, and gender. To test against this baseline, we also include as a feature the speech-level output of each of the four previous models for NRA rating classification. To ensure that the features applied in these models are independent, we plotted features' correlations against each other in Figure 2, and see expected results of fairly low correlation among all pairwise comparisons.

The Multinomial Logistic Regression is given by $ln \frac{\pi_j}{\pi_J} = \alpha_j + \beta_j X$, where the outcome is classified into, e.g., $J = 4$ categories indexed by j, and $\pi_j = \frac{1}{1+\sum e^{(\alpha_j + \beta_j X)}}$.

For all five machine learning algorithms described, we use k-fold cross validation to assess how well a model can be generalized to an independent dataset. We decided our dataset into 5 folds, each time holding out one of the folds as development set and the others as training set. As a result, we train k times and get k errors. The model with the lowest error is taken and applied to our test set.

5 Results

5.1 Experiments & Results

We tuned our models using 5-fold cross validation. This was most relevant for determining which set of features to use. We tested the following ranges of word frequencies to use for our featurization: $(1000, 4000)$, $(100, 4000)$, $(300, 4000)$, $(500, 20000)$, $(500, 3600)$, and $(500, 4000)$. Out of each of these possible feature frequency ranges, cross-validation across multiple models consistently showed that the $(100, 4000)$ range performed the best. (Although the $(500, 20000)$ range seems very large,

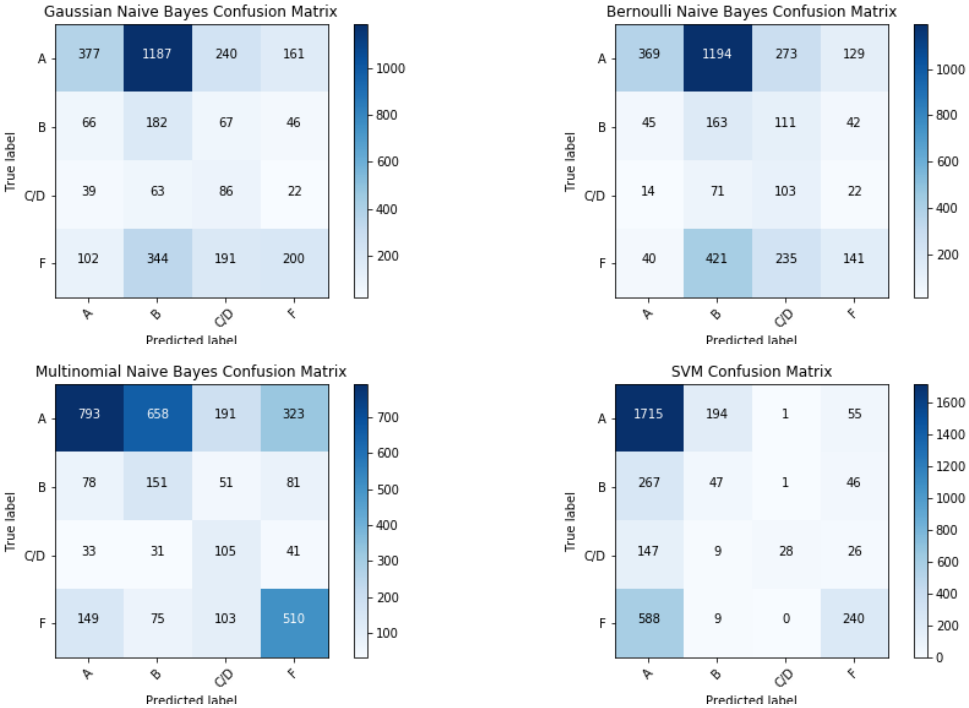

Figure 3: Confusion matrices for the four methods considered

there are few words that occur with a frequency between 4000 and 20,000, and those that do are too frequent to be very informative).

We used accuracy as our primary metric to determine the success of our algorithm. Looking at the confusion matrices was also helpful in determining potential problems with algorithms. For example, in Figure 3 we can see that both Gaussian Naive Bayes and Bernoulli Naive Bayes make completely inaccurate predictions. Looking at the confusion matrix we can see that the SVM method, while achieving high accuracy, seems to classify almost everything as an "A" rating, which achieves high accuracy purely based on the fact that "A" is the most common grade of a congressperson in our dataset. However, the confusion matrix for Multinomial Naive Bayes indicates that the algorithm is picking up on subtler aspects of the data because not only are many "A" speeches classified correctly, but many "F" speeches are as well. In fact, the diagonal is the largest entry in each row for this confusion matrix, which means that each speech was more likely to be categorized as its true label than any other individual label. This confusion matrix also led us to explore a three-category classification where "A" and "B" are joined into one class, because "A" speeches were being misclassified as "B" speeches. This not only decreased the error in every case (which is what we would expect, because fewer categories means that it is easier to guess right), but the test error was even lower than the training error in two cases.

To gain greater insight into the signals Multinomial Naive Bayes receives, we looked at the words that were most predictive of each class. The four most predictive (stemmed) words for an "A" rating were "obamacar", "kentucki", "growth", and "promis". The four most predictive words for an "F" rating were "gentleman", "bank", "climat", and "gun". The "A" rating words seem to be related to the Republican party (Obamacare is the name given to President Obama's healthcare bill by the Republican Party, Kentucky is a famously Republican state, etc.), while the "F" words are more related to Democratic concerns (Democrats traditionally speak out more about climate change, bank regulation, and gun control). The results for all of our approaches are available in Table 2.

5.2 Discussion

The Multinomial Naive Bayes and SVM algorithms perform the best on our data. The former performing well (with test error of 53.8%) indicates the number of occurrences of a particular word is important (in contrast to, e.g., the Bernoulli model that only considers whether a word is present,

Table 2: Error Comparisons Across Models Using 20% Test Sets, 4- and 3-Classification Levels, and Frequency Featurization

Models	Features	4-Class Training Error (%)	4-Class Test Error (%)	3-Class Training Error (%)	3-Class Test Error (%)
Multinomial Regression	Party, State, Age, Gender	4.0	3.7	3.1	3.1
Gaussian Naive Bayes	Featurized Text	60.2	74.9	68.4	64.2
Bernoulli Naive Bayes	Featurized Text	64.3	77.0	41.9	44.4
Multinomial Naive Bayes	Featurized Text	38.9	53.8	37.7	42.0
Support Vector Machine	Featurized Text	48.9	39.8	32.0	27.1
Multinomial Regression & Multinomial Naive Bayes	Party, State, Age, Gender, & Featurized Text	4.6	6.2	2.4	4.4

and has a higher test error of 77.0%. The SVM algorithm performed well (with test error of 39.8%) because it creates an explicit separation of the training set and does not assume independence of features. As expected, Gaussian Naive Bayes had high test error (74.9%) because out data are discrete and not continuous. On the whole, none of the text-classification algorithms performed well due to the quality of input data. Very few speeches mention gun rights specifically, and the model is likely prone to make a Democrat vs. Republican classification from the text, which will be less accurate than simply using the actual party affiliation feature of each congressperson. We note that including speech data predictions of NRA ratings as features in the regressions alongside other politician features yields a lower accuracy by 1 to 3 percentage points. However, without speech data, multinomial logistic regressions on politician features alone are over 95% accurate because party is a high predictor of NRA rating. The training and test error are quite similar in this case, so we do not believe we are overfitting; regardless, when only using party and state features as indicator variables (specifically to decrease dimensionality to avoid overfitting), we still have over over 90% accuracy.

One application of our results is the ability to predict NRA ratings with 60.2% accuracy for up-and-coming politicians who have made speeches. For example, based on our Multinomial Naive Bayes model, we predict Doug Jones, an unrated candidate for Senator of Alabama, to have rating C/D from the NRA [7]. This rating potentially makes sense because Doug Jones is a gun-friendly Democrat, indicating that he might get a mid-level rating from the NRA. Results such as this can be extrapolated to intuit how much campaign funding future politicians might receive from the NRA.

6 Conclusion & Future Work

Overall, our research shows that basic congressperson features such as party affiliation, state, age, and gender are highly accurate predictors of NRA rating. Speech analysis alone reaches at best about a 60% accuracy when using Multinomial Naive Bayes or SVM. These models function the best for text classification using our data because of multiple word occurrences are significant, and and lack of word independence assumption, respectively. Further, when including the speech predictions of NRA rating as a feature in the regression model alongside congressperson features, the model accuracy decreases due to the increased noise from the inaccurate speech models.

To improve our performance we could use word embeddings to featurize the speeches, which will our models to incorporate some sentiment analysis. Further, we could implement `DFreq` to featurize based on word counts across speech documents, rather than analyzing the count of words within a document. This method could be more informative because it would capture words that do not show up in all speeches but have a reasonably high frequency. Lastly, more work can be done on determining how to optimally categorize ratings into bins in order to yield more predictive power.

7 Contributions

Anqi Ji fully implemented both Naive Bayes and SVM methods in Python on all different versions of featurizations, number of classifications, and test / training splits and performed the literature review.

Allison Koenecke performed data scraping, preliminary top-line checks, and cleaning of the NRA data downloaded. She wrote the code for k-fold validation and test / training dataset splitting. She further performed all regression analyses involving baseline metrics (i.e., including congressperson attributes as features in addition to or without speech data).

Julia Olivieri downloaded, cleaned, and automated several featurization methods for all speech data. She also wrote the code to merge the two data sources and generate test / training datasets. She further ran top-line checks of the speech data, tested different featurization methods for robustness, and refactored the Naive Bayes and SVM learning algorithms.

All team members contributed equally to the writing of the final presentation and paper.

References

[1] Discover u.s. government information. `https://www.govinfo.gov/`. Accessed: 2017-11-21.

[2] Feature selection using improved mutual information for text classification.

[3] Gaussian naive bayes. `http://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes`. Accessed: 2017-11-21.

[4] A nation divided: Classifying presidential speeches. `http://cs229.stanford.edu/proj2016/report/AcharyaCrawfordMaduabum-ClassifyingPresidentialSpeeches-report.pdf`. Accessed: 2017-10-19.

[5] Where congress stands on guns. `https://projects.propublica.org/guns/#nra`. Accessed: 2017-11-21.

[6] G. Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, Mar. 2003.

[7] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 338–345, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[8] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48. AAAI Press, 1998.

[9] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226–1238, 2005.

[10] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recogn. Lett.*, 15(11):1119–1125, Nov. 1994.

[11] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 42–49, New York, NY, USA, 1999. ACM.