STAT 195 Final Report

Peter Shen & Harry Burke

Introduction

Predicting the outcome of an election is fraught with difficulty. Not only is an individual's vote based on many competing, often heterogeneous factors, but the outcome of an election can shift over time as public sentiment changes. Further, many voting decisions are made on more qualitative grounds, that is, individuals often vote for candidates they feel a personal connection to, or whose opinions align with theirs. However, we will show that although voting is a highly personal and idiosyncratic process, we can predict the outcome of the 2018 midterm election using only quantitative variables corresponding to factors such as amount of money raised and incumbency status.

There are a few important issues to consider when trying to predict the outcome of an election. First, there is often a lack of data regarding election results. Because elections only occur every two years, obtaining more than a few elections worth of samples can be challenging. Further, data quality may decrease in past elections, leading to sparsity and noise. To solve these problems we need a machine learning method that is not too sensitive to noise, and we may need to impute missing data. Further compounding this problem, we often choose to examine a large number of variables, as we might think that there might be a multitude of factors that go into a person's choice of candidate. Because of this, we need a machine learning method that can handle having more parameters (features) than samples. Lastly, while we might like to think that our features are each independently informative, this assumption is often very wrong; related features tend to be very highly correlated, such as the amount of money a candidate receives in donations and their polling popularity. Because of this, we need a machine learning method that can handle having blocks of correlated features.

Naively, we might think a good idea is to use a generalized linear model with a logistic link function, as the linear regression model, under certain assumptions, is the best linear unbiased estimator, that is, it produces the lowest variance estimate. (Chipman, 2011) However, there are a few issues with this. First, Zou and Hastie claim that the Ordinal Least Squares estimate for the linear model is often a poor choice for prediction. (Zou & Hastie, 2005) Further, as the number of parameters in the model increases beyond the number of samples, it becomes easy to overfit. OLS also has no way to deal with highly correlated parameters. In our preliminary prediction we chose to tackle these problems with partial least squares, however it had a few downsides. Partial least squares works by projecting the predictor and outcome variables into a new space and fitting a linear regression model. The most important drawback was a lack of interpretability. Because partial least squares involves a linear transformation of the features, it does not easily lend itself to interpretation. For our final model we instead chose to use elastic-net.

Elastic net, put simply, is a combination of L1 (LASSO) and L2 (Ridge) regularization. Both L1 and L2 norms are penalties put on the resulting regression model to try to solve the problem of having too many parameters, as well as correlation in the data. The L1 penalty, used in LASSO regularization, performs both shrinkage and variable selection on the data by driving some coefficients to zero. (Zou and Hastie, 2005) However, Zou and Hastie list a few limitations make it unsuited for our analysis. First, if you have more parameters than samples, LASSO typically constrains you to typically having at most n parameters in the final model. Further, if you have blocks of correlated parameters, LASSO selects one from the block at random. The solution to these problems is to add the L2 penalty, which creates Elastic Net. We want the ability to set parameters to zero that LASSO has, but we also want to be able to deal with highly correlated variables, as we suspect that a lot of our variables are related to one another. Elastic Net has some of the advantages of both L1 and L2 regularization, that is, it performs variable selection, but it also deals with correlation among the features well. We used alpha = 0.5 to treat correlated variables as groups.

Results

Coefficients in elastic-net. We found that elastic-net did not shrink any coefficients to zero (table 1), but rather each coefficient still contributed at small levels of lambda and alpha. Further, incumbency status was by far the strongest predictor, with a beta coefficient over ten times larger than the next largest beta. Our predictions suggest that there is a strong momentum behind the incumbents as a candidate. Interestingly, approval rate and seat transition were found to have the largest negative beta coefficients, which implies that the lower the presidential approval, the more likely a house seat was to change to another party.

Table 1: Coefficient found by elastic-net at Lambda found using the “one-standard-error” rule.

Feature	**Coefficient**
Incumbent1	4.93648715
minority	0.40070118
vote	0.35969054
share	0.07361817
primary_vote	0.02403664
labor	0.0041163
income	1.46E-06

raised	2.70E-07
name	-0.0002987
bachelor	-0.0036369
seat_transition4	-0.0086342
gov_party1	-0.2181773
same1	-0.2309611
seat_transition1	-0.3175348
seat_transition3	-0.3769551
gender1	-0.4771441
approval	-0.5438488
(Intercept)	-3.8525022

We show in figure 1 and figure 2, the higher the penalty, the more the variates shrink towards 0. Interestingly, the lowest binomial variance is found when all variates are added in the model.

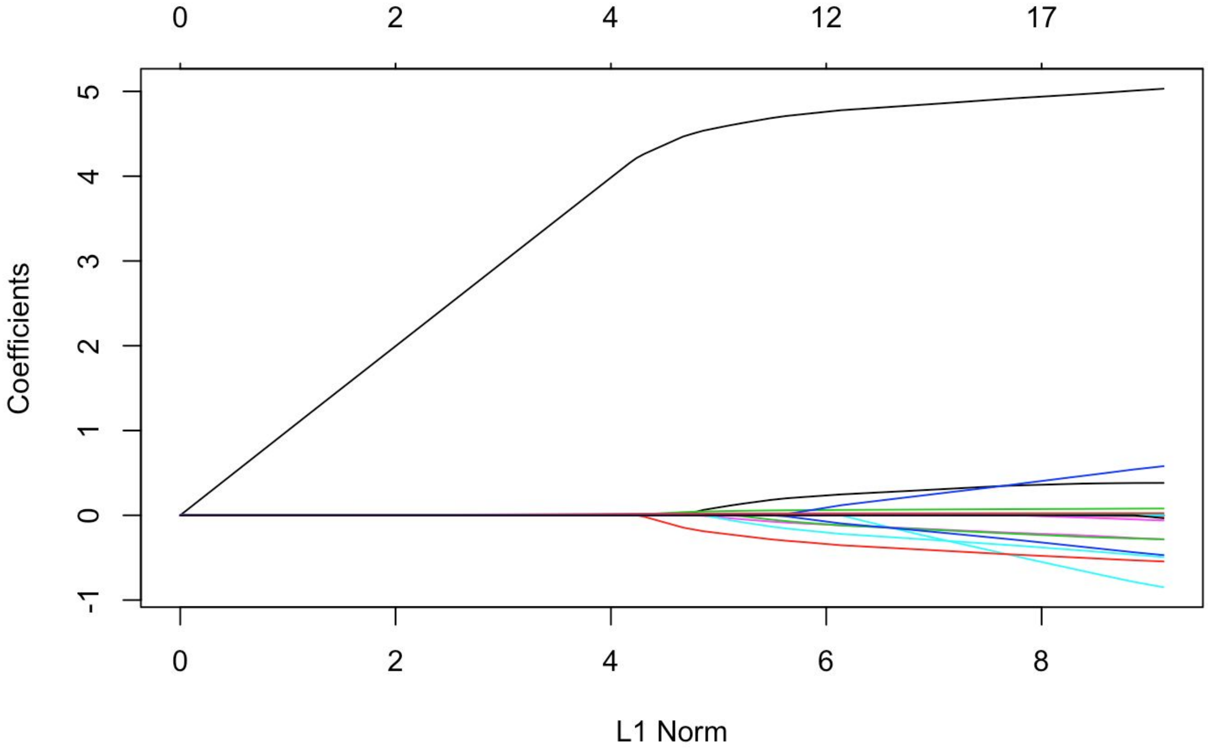

Figure 1. Values of model coefficients as L1 Norm and alpha increase.

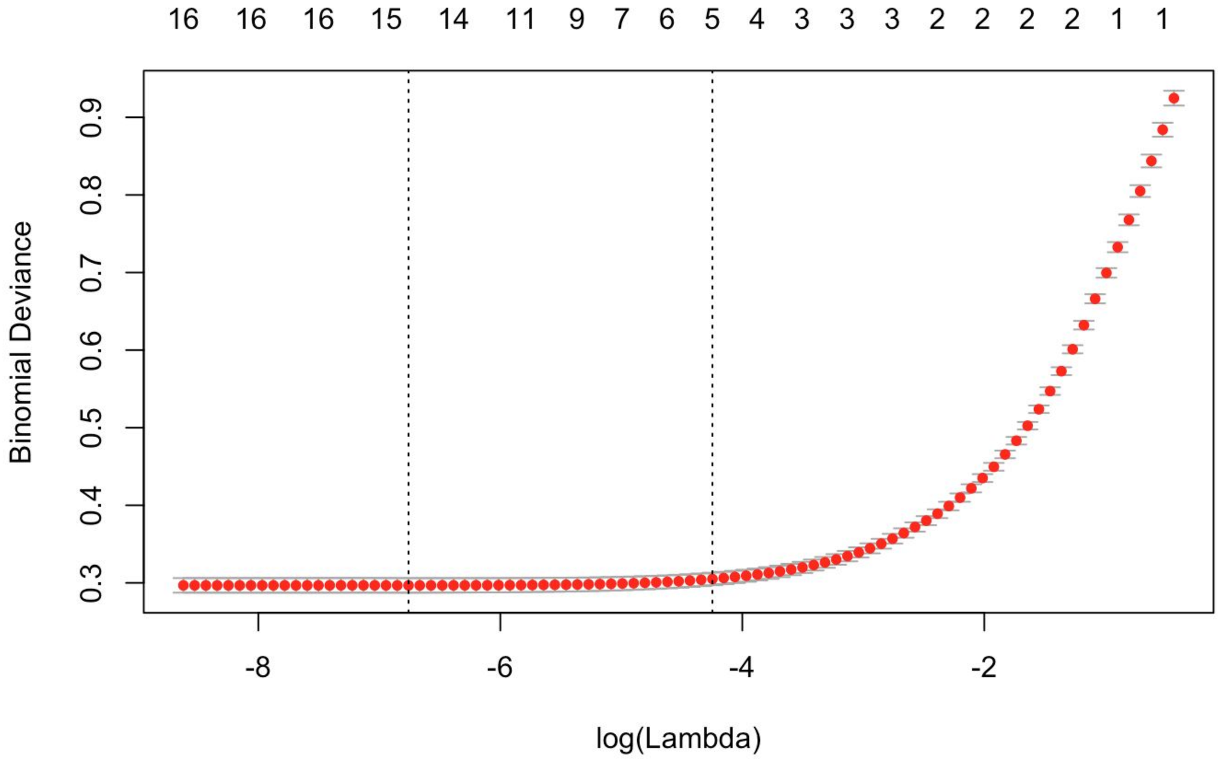

Figure 2. Binomial deviance -- the error rate of the logistic regression prediction -- increases as Lambda increases. Low contribution coefficients are shrunk and dropped out as Lambda increases.

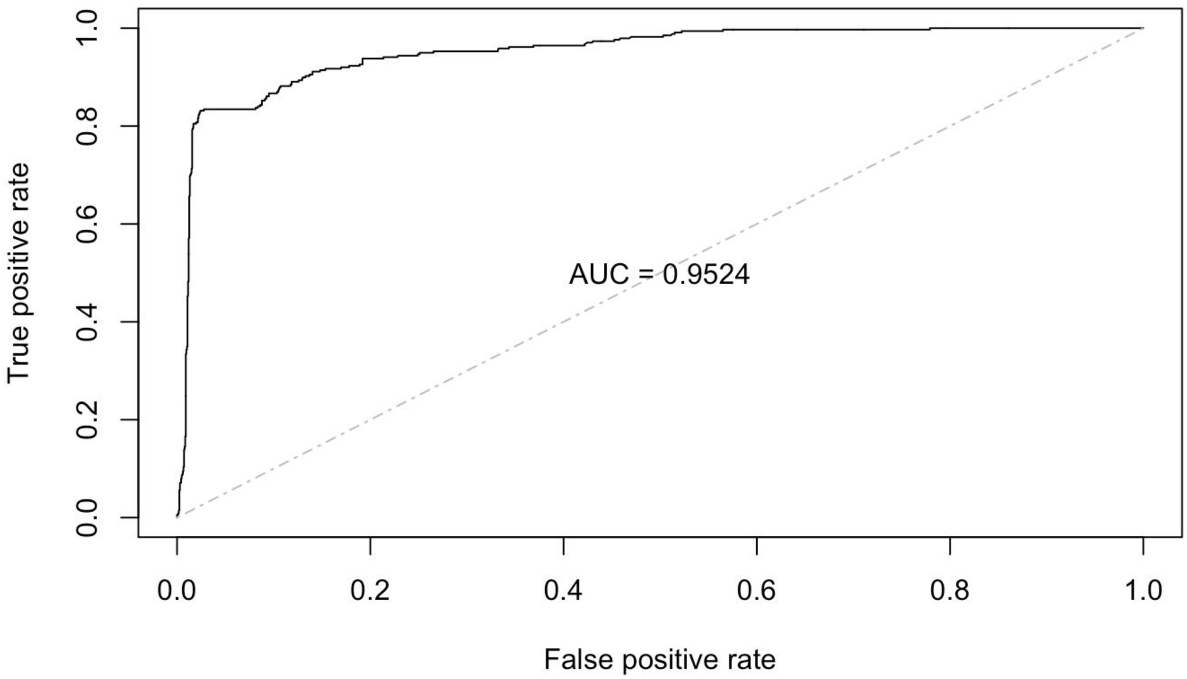

Figure 3. ROC for the test set containing 20% of the randomly sampled data from the 2010 to 2016 dataset. We scored an AUC of 0.9524.

Figure 3 shows a measure of test accuracy, the ROC. With and ROC of over .95, we found our model was highly predictive of election outcomes in test data.

Table 2. Prediction of house majorities

Party	**Probability of Majority**
Republican	88.62%
Democrat	11.38%

Lastly, we calculated the probability of a democrat majority, and a republican majority following the 2018 election. We found an **88% chance of republican majority** and a **12% chance of a democratic majority**, suggesting there is a low likelihood of democrats retaking the house following this midterm election.

<u>Methods</u>

Using the logistic regression model and elastic-net. We are modelling the voting outcomes as either 0 for Democratic or 1 for Republican, which for a prediction for 1/Republican is given by:

$$Pr(G = 1 \mid X = x) = \frac{e^{\beta_0+\beta^T x}}{1+e^{\beta_0+\beta^T x}}$$

The objective function is the penalized logistic regression which is minimizing negative binomial log-likelihood

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} -\left[\frac{1}{N}\sum_{i=1}^{N} y_i\cdot(\beta_0+x_i^T\beta)-\log(1+e^{(\beta_0+x_i^T\beta)})\right]+\lambda\big[(1-\alpha)||\beta||_2^2/2+\alpha||\beta||_1\big]$$

In the new forecast, we decided to use multiple linear regression with the elastic-net regularization penalty, where we control for groups of features by setting α equal to 0.5 in the negative binomial log-likelihood. We suspect that there are correlations between features, for example, amount of money raised and the amount of money spent, the percentage of minorities and median income, so L1 penalty would have issues with feature selection and behave more erratically. We also had a lot of sparsity in our data because features such as gender, incumbency status, and seat transitions are categorical values, which are converted to categorical encodings, so L2 penalty would control and stabilize these features by introducing convexity. Under these assumptions, we would like to have a mix of both L1 and L2 penalties, and therefore we use the elastic-net regularization to control for both penalties. Although there is a drawback in terms of increased computational time for elastic-net, due to optimizing both α and λ, given the small dataset (less than 10,000 samples) this is not a major concern.

Imputing missing values in the data. Since the early forecast, we changed our imputation of missing values to be more accurate. For the amount of money raised and spent we set missing values to 0 because imputation will likely not give reliable values, because we believe that non-missing values would tend to be much higher than missing values, and that most missing values for money raised implies a low amount of money. In the early forecast, we performed hot deck imputation (sampling from the observed distribution) for values of `income` and `approval rates.` However, in the new forecast, we imputed the values for `income` and `approval rates` using the mean `income` or `approval rate` values within the year, same state, and same party for the candidate since these values are likely to be closer to the true values.

Model validation using cross-validation and test set. One of the assumptions we made early on was to assume that each candidate running each year is independent and identically distributed (IID). We realize that this may not be a true assumption since the geographical relationships and outcomes can be closely related between candidates. However, we can still randomly sample from the dataset to split into training and test datasets, so long as we we sample within year and state. We split our initial 9998 samples into 7998 training samples and

2000 test samples. We used 10 fold cross-validation for feature selection on the training sample, and then validated our fitted function on the test set.

Prediction of 2018 election results. After cross-fold validation, we used the "one-standard-error" rule (Hastie, T., Tibshirani, R., & Friedman, J., 2001) where we pick the most parsimonious model within one standard error of the minimum and is a more a conservative approach. We then predict the 2018 election results using the coefficients selected from elastic-net in a linear regression model.

Prediction of majority seats. In order to predict which party will gain a majority of the seats in the house, we need to consider the probability of winning more than half of the seats. For this, we need to have some sort of joint probability for the democratic and republican winners for each seat. We simplified our approach by not considering any other parties besides democrats and republicans, since another party gaining majority of the seats is very unlikely. For each seat, we considered the democratic and republican candidates with the highest probability of winning that seat, and then normalized for their probabilities between 0 and 1 since there were cases where the sum of the probabilities could be greater than 1. We then calculated the right-tailed Poisson Binomial distribution density of greater than half using the probabilities of each party individually. We chose the Poisson Binomial distribution because we are able to create a distribution density using individual probabilities for each event -- in this case a seat, compared to the binomial distribution where the probabilities are constrained to be constant for each event.

Code

Code available at: https://github.com/p-shen/2018-midterms-prediction

References

Chipman, J.S., 2011. Gauss-markov theorem. In *International Encyclopedia of Statistical Science* (pp. 577-582). Springer, Berlin, Heidelberg.

Friedman, J., Hastie, T. and Tibshirani, R., 2001. *The elements of statistical learning* (Vol. 1, No. 10). New York, NY, USA:: Springer series in statistics.

Zou, H. and Hastie, T., 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), pp.301-320.