Predicting Votes from Census Data

Tynan Challenor SUNet ID: tynan

December 15, 2017

1 Introduction

1.1 Motivations and project description

Predicting voter turnout has been a persistent obstacle for campaigns and analysts trying to model election outcomes [1]. The stakes for accurately predicting which parts of the electorate will vote are different depending on the interest group: campaigns, for example, have only a limited number of people and dollars at their disposal. Efforts to increase voter turnout can include setting up phone banks, organizing volunteers to pass out fliers, and recording robo-calls [2]. Understanding which voters are least likely to vote can guide a campaigns use of money and person-power.

In the Pew Research Centers analysis of the 2014 congressional elections, they found that having more accurate likely voter models would have flipped their predictions [1]. This projects goal is to begin the search for useful, publically available data that can be used to build a robust likely voter model.

After the 2016 presidential election, the U.S. census bureau sent an addendum to its labor force and demographics survey consisting of 152,095 respondents. The addendum asked whether each respondent had voted in the election [3]. The original survey included 374 questions, including follow-up probes depending upon the answers provided. The questions mostly focused on where respondents fit into the labor force, including queries such as, *Last week could you have started a job if one had been offered?* These 374 questions were used as features, and each respondent as an example to train three models: a support vector machine, a naive bayes predictor, and a logistic regression predictor. Each model was asked to predict *yes or no* whether an individual voted in the election based upon their responses to the survey. Furthermore, kmeans was used to cluster the data by features in the hopes of finding sub-sets of responses that might be useful for predicting ones propensity to vote. Logistic regression coefficients with the greatest positive and negative values were matched with their corresponding survey questions.

In addition to creating a model that accurately predicts whether someone will vote, or returns true probabilities, it will also be important to try to understand the semantic meaning behind what the model is learning. Census bureau data is not available for every citizen in the United States, so in order to make predictions on other potential voters for future campaigns it will be important to know which information is most essential to collect prior to predicting.

1.2 Related work

There is little work being done right now to explore using machine learning to explicitly predict whether individuals will vote. A number of studies have used natural language processing with social media, such as twitter, to do sentiment analysis monitoring relative support for candidates around an election. For example, Birmingham and Smeaton (2011) combined sentiment analysis of tweets with polling data of the Irish General Election in 2011, but did not return election predictions that were more accurate than traditional polling methods [4]. Their work differs from mine in that my goal is improve upon one part of election predicting, assessing likelihood of voting, rather than to generate a prediction of the election itself. Bollen et al. 2011 used twitter to model national mood in the U.S. [5]. They created a vector with six emotional features such as depression, vigor, and confusion, and monitored how national sentiment changed after large, country-wide events such as elections or major changes in the economy. They found that they could see mood swings after larger national events and posit that they might be able to predict such events by first looking at national mood. Again, my own work is different because it aims to delve much deeper than a broad trend to ascertain who on an individual basis is likely or not likely to vote.

The data journalism site 538.org uses polling data and prior voting records to model current elections [6]. While their statistical predictions are often accurate and they make use of likely voter models to guide their own work, they as well are focused more on the question of accurately predicting an election, rather than accurately predicting who will vote.

Prior methods used to build likely voter models were based on targeted survey questions. For example, seven questions developed by Paul Perry of Gallup in the 1950s and 60s have been subsequently adopted by Gallup and Pew

to build their models [1]. One question, for example, reads: *How much thought have you given to the coming November election? Quite a lot, some, only a little, none.* The Pew Center took responses to those seven questions and added demographic factors such as age, education, race, party affiliation, home ownership, and length of residence at current home; they also added a feature that accounts for an individuals past voting record using old voter roles. With this feature set they trained a logistic regression model and used random forests to get two different probabilities of voting for each individual. The coefficients from logistic regression were also used to try to identify the most predictive features so that a similar model could be built for an election with different specifications presidential versus congressional, 2018 versus 2014 for example.

2 Dataset and features

The dataset that I used came from a 2016 census bureau survey focusing on labor force demographics [3]. It contained responses from 152,095 individuals, of which I used 75,000 for training my models due to storage constraints. Labels indicating whether an individual voted or not came from an addendum to the survey sent out after the election inquiring about whether respondents had voted and reasons for not doing so. I used the answers to the first of these questions as my labels; positive responses were labeled 1, everything else (*no, dont know, refused to respond*) was recorded as 0 in pre-processing. All of the other questions from the addendum were removed from the feature set, such as *Which of the following was the main reason you were not registered to vote*, a follow up from the previous question that gives away whether the respondent voted or not.

The data were normalized to have mean zero and unit variance. They were divided into train, validation, and test sets using 64% for training, 16% for validation, and 20% for testing. Some of the questions and responses are as follows:

Table 1: Sample of features from census bureau data

Question	Entry
Metropolitan status	(1) Metropolitan (2) Non-metropolitan (3) Not identified
Type of discouraged worker	(1) Discouraged worker (2) Conditionally interested (3) Not available
Last week, could you have started a job had one been offered?	(1) Yes (2) No

3 Methods

3.1 Logistic Regression

Logistic regression is a natural algorithm to use for predicting likelihood of voting. It is a well-calibrated model, meaning that the probability returned by the model for a given training example is similar to the proportion of training examples that actually have that label. For example, if there were 10 training examples that the model assigned a 33% chance of voting, then we'd expect approximately three of those 10 people to have actually voted.

Logistic regression is a classifying algorithm with hypothesis function given by:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

In the equation X is the matrix of inputs, each row being another example data point from the survey and each column being another question in the survey. The vector theta is the set of parameters learned by the model. The function g is the sigmoid function, and returns values between 0 and 1. Predictions above a certain threshold will be labeled class 1 and values below that threshold are labeled class 0. Because logistic regression returns values between 0 and 1, its outputs are particularly well suited to being interpreted as probabilities. To optimize the hypothesis function we take the derivative of the log likelihood of the parameters, θ, with respect to θ and perform gradient ascent. The log likelihood is expressed like this:

$$l(\theta) = \sum_{i=1}^{m} y^{(i)} log(h(x^{(i)})) + (1 - y^{(i)}) log(1 - h(x^{(i)}))$$

and the gradient ascent update is given by:

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

3.2 Support Vector Machine

A support vector machine was also trained using the census data. The theory behind an SVM involves finding the maximal geometric margin, which can be thought of as the gap that separates positive predictions from negative ones. The larger the gap, the greater the confidence of the model. Each prediction can be found by solving the following optimization:

$$w^T x + b = (\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)})^T x + b \tag{1}$$

$$= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b \tag{2}$$

where the α's are Lagrange multipliers. There are a couple of key points behind these equations: first, with the exception of the support vectors, the alphas will be zero, which simplifies the sum. Second, prediction can be written as the inner product $\langle x^{(i)}, x \rangle$, which becomes useful if using the kernel trick for problems in large (up to infinitely high) feature space.

3.3 Naive Bayes

The last classifier tested was Naive Bayes, which works by trying to predict the likelihood of seeing the feature vector, x, given a label, y. The naive assumption underlying the name is that all of the features in x are conditionally independent given y, which simplifies the calculation down to:

$$p(x_1, ..., x_n \mid y) = p(x_1 \mid y)p(x_2 \mid y, x_1)...p(x_n \mid y, x_1, ..., x_n) \tag{3}$$

$$= p(x_1 \mid y)p(x_2 \mid y)...p(x_n \mid y) \tag{4}$$

$$= \prod_{i=1}^{n} p(x_i \mid y) \tag{5}$$

Finally, using Bayes rule, the model predicts a label for unseen data using the equation

$$p(y = 1 \mid x) = \frac{p(x \mid y = 1)p(y = 1)}{p(x)} \tag{6}$$

$$= \frac{(\prod_{i=1}^{n} p(x_i \mid y = 1))p(y = 1)}{(\prod_{i=1}^{n} p(x_i \mid y = 1))p(y = 1) + (\prod_{i=1}^{n} p(x_i \mid y = 0))p(y = 0)} \tag{7}$$

3.4 K-Means

K-means is an unsupervised clustering algorithm. The algorithm initializes k centroids and then clusters the data by placing examples into a cluster based on the minimum distance between a sample and a centroid. It then updates the centroids to be the mean of the examples in that cluster and runs the algorithm again until convergence. The algorithm is useful for identifying unknown patterns in the data. One downfall of kmeans might be that the algorithm converges in a local minimum. This can be avoided by running the algorithm multiple times and comparing outputs from the distortion function, given by:

$$\sum_{i=1}^{m} || \, x^{(i)} - \mu_{c^{(i)}} \, ||^2$$

where $\mu_{c^{(i)}}$ is a given centroid.

3.5 Principal Component Analysis

Principal component analysis (PCA) is a way of transforming data from an n-dimensional feature space to one of k-dimensions where k < n. PCA is useful if there is reason to believe that some of the features in the input are highly correlated and it would therefore be more effective to train a model using a smaller feature space. It is also a useful tool for visualizing data because it can collapse an n-dimensional vector into a two dimensional one; this is the context

that I use in this project. The conceit is to project the vectors x onto a unit vector u; the direction of u is selected so as to maximize the variance of the projected data. Maximizing the variance is the same as maximizing the following:

$$u^T(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}x^{(i)^T})u$$

Solving the above given the constraint that the $|| u ||^2 = 1$ results in the principal eigenvector of Σ, the empirical covariance matrix of the data.

Results and Discussion

Because the task is one of classification, I'll evaluate the performance of each method using the summary statistics recall, precision, and accuracy. Recall, also known as the true positive rate, can be thought of like this: of the people who actually voted, what fraction did the model predict voted? Precision, by contrast, describes how selective the model is when assigning its labels. If the model predicted that everyone voted, it would have perfect recall, but a 50% precision. Lastly accuracy describes how many people the model labeled correctly (either for voting or not voting) out of all of the examples. We can create a confusion matrix using the true negative count (tn), false positive count (fp), false negative count (fn), and true positive count (tn). Summary statistics are defined thus:

$$Recall = \frac{tp}{tp+fn} \quad Precision = \frac{tp}{tp+fp} \quad Accuracy = \frac{tp+tn}{tp+tn+fp+fn}$$

The confusion matrices for the models tested are as follows:

Table 2: Logistic regression confusion matrix

$total = 15001$	Predicted did not vote	Predicted did vote	
Actually did not vote	tn = 7268	fp = 1208	Total: 8476
Actually did vote	fn = 619	tp = 5906	Total: 6525
	Total: 7887	Total: 7114	

Table 3: Support vector machine confusion matrix

$total = 15001$	Predicted did not vote	Predicted did vote	
Actually did not vote	tn = 7128	fp = 1348	Total: 8476
Actually did vote	fn = 587	tp = 5938	Total: 6525
	Total: 7715	Total: 7286	

Table 4: Naive Bayes confusion matrix

$total = 15001$	Predicted did not vote	Predicted did vote	
Actually did not vote	tn = 5011	fp = 3465	Total: 8476
Actually did vote	fn = 150	tp = 6375	Total: 6525
	Total: 5161	Total: 9840	

Table 5: Performance of all models

	Recall	Precision	Accuracy
Logistic regression	90.51%	85.75%	87.82%
Support vector machine	91.00%	84.10%	87.10%
Naive Bayes	97.70%	59.12%	75.90%

Both logistic regression and the support vector machine performed with similar accuracy and recall. In analyzing the models it is important to keep in mind the two practical applications of the project: the first is finding a model that will make quality predictions given relevant data. For example, if a political campaign had up-to-date census bureau data on a group of potential voters, then my results indicate that both logistic regression and an SVM will perform moderately well. However if a campaign were looking to distill which features were the most important, or the probability that a certain individual (or group of individuals with a similar demographic fingerprint) will vote, then logistic regression becomes a more valuable tool.

First we can look at the coefficients returned by logistic regression to see which features were most highly correlated with voting.

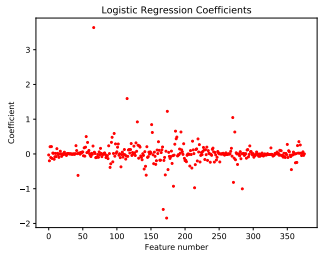

Figure 1: Plot of the coefficients returned by logistic regression. While many are close to zero, there are a few positive and negative coefficients who's features are semantically meaningful.

Table 6: Features most correlated with voting

Coefficient	Question	Possible responses
3.64	Highest level of school completed or degree received	$> 1^{st}$ grade, 2nd-4th,...GED, Bachelors etc.
1.59	Marital status	Married, divorced, separated etc.
1.23	Which *major* industry do you work in?	Agriculture, mining, construction, wholesale, etc.

Table 7: Features least correlated with voting

Coefficient	Question	Possible responses
-1.84	Marital status based on armed forces participation	Married civilian spouse, divorced, etc.
-1.60	Which *intermediate* industry do you work in?	Agriculture, mining, arts, information, etc.
-1.00	Person's age	Age

Which sector of the labor force an individual occupied (major and intermediate) and marital status were both strong indicators of one's propensity to vote or not vote. Education level was a the highest determinant of voting given this data, which is something that others in data journalism have already pointed out [7]. However, what these data suggest is that semantically meaningful data, such as one's education, age, job or marital status can be combined with large scale data to create a model that correctly predicts close to nine out of ten times whether an individual will vote. The majority of questions in the labor force survey that I used are detailed inquiries into how much one works and and the methods a respondent might be using to look for a job if in need of one. While each of these features by itself may not be useful for predicting voting, the sheer quantity of features and examples helps to create a fingerprint that might prove useful for future work in assessing likely voter pools.

Unfortunately clustering the data by feature and then using PCA to shrink the samples to two dimensions yielded no clear patterns. Individuals in each cluster were equally likely to vote:

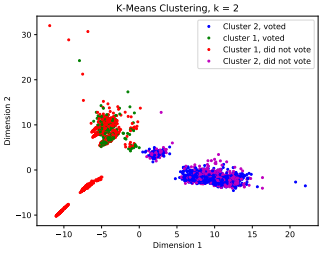

Figure 2: Input data clustered into two groups (red/green and blue/purple) with blue/green indicating voting and purple/red indicating non voting. Voting and non-voting are both distributed quite evenly over the clusters.

Conclusion

Training an SVM and a logistic regression model using labor force data from the U.S. census bureau predicts whether an individual will vote with approximately 87% accuracy and 90% recall. Future work to find relationships between the features included in large, sprawling datasets might help pollsters and campaigns discover the most salient indicators of voting. A true likely voter model would also need a strategy for modeling elections that take place in different years. We don't yet know whether turnout for a 2016 presidential election might provide good training data for a 2018 congressional mid-term, but that would be a worthy next step.

References

[1] R. Igielnik, and S. Keeter 20036 U.-419-4300 | M.-419-4349 | F.-419-4372
| M. Inquiries, Can Likely Voter Models Be Improved?, Pew Research Center, 07-Jan-2016.

[2] D. P. Green and A. S. Gerber, Get Out the Vote: How to Increase Voter
Turnout. Brookings Institution Press, 2015.

[3] Current Population Survey FTP Page. [Online]. Available:
https://thedataweb.rm.census.gov/ftp/cps_ftp.html#cpssupps. [Accessed: 14-Dec-2017].

[4] A. Bermingham and A. Smeaton, On using Twitter to monitor political
sentiment and predict election results, in Proceedings of the Workshop on
Sentiment Analysis where AI meets Psychology (SAAIP 2011), 2011, pp. 210.

[5] J. Bollen, H. Mao, and A. Pepe, Modeling public mood and emotion:
Twitter sentiment and socio-economic phenomena., ICWSM, vol. 11, pp. 450453, 2011.

[6] N. Silver, How The FiveThirtyEight Senate Forecast Model Works,
FiveThirtyEight, 17-Sep-2014. .

[7] N. Silver, Education, Not Income, Predicted Who Would Vote For Trump,
FiveThirtyEight, 22-Nov-2016.

[8] I used Scikitlearn to train logistic regression, SVM, and naive bayes. The developers of scikit learn:
Fabian Pedregosa, Gal Varoquaux, Alexandre Gramfort, Vincent Michel,
Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron
Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau,
Matthieu Brucher, Matthieu Perrot, douard Duchesnay; 12(Oct):28252830,
2011.

I would also like to thank and acknowledge my friend and old roommate,
Michael Xie, who loves machine learning and is always willing to chat about
how things are going and offer his two cents.