Using Poisson Binomial GLMs to Model Voter Behavior

Evan Rosenman
Stanford University
rosenman@stanford.edu

Nitin Viswanathan
Stanford University
nviswana@stanford.edu

1. Abstract

2. Introduction

Political organizations seek to build voting models to explain voter preferences, as this information is vital for voter targeting. In the US, voting data indicating how many votes every candidate receives is available at the precinct level, but the most granular data who voted for which candidate is private. As a result, political groups are forced to rely on polling data to perform analysis at the individual voter level. Polling data can not only be unreliable but it is also incomplete, as voters are not obligated to express their true preferences and the majority of voters will not participate in polls. We develop individual voter models based off of publicly available aggregated voting data.

We use precinct-level data across several counties in Pennsylvania from the 2016 presidential election together with individual-level demographic data to model who a given voter will vote for. We focus on predicting votes for both Hillary Clinton and Donald Trump in the state of Pennsylvania as it was a key swing state that went in favor of Donald Trump in the 2016 presidential election. Trump's margin of victory was very slim at only 44,292 votes, only 0.72% more than Clinton's[13]. Precinct-level election results are the most granular results available; for reference there are about 9,000 precincts total in Pennsylvania [13].

We formulate the problem using a Poisson binomial generalized linear model and maximize the log-likelihood, which has not, to our knowledge, been done before in other literature.

Our paper makes two key contributions:

- We develop the math necessary to use and train Poisson binomial GLMs
- We apply poission binomial GLMs to the specific task of predicting voter behavior

3. Related Work

Theoretical work on the Poisson Binomial distribution has focused on computationally tractable ways to estimate its distribution function, often via approximations to other distributions [6, 10, 3]. Prior research [7] has identified a closed-form expression for the CDF, which relies on the discrete Fourier Transform. This technique is leveraged in the `poibin` package [12], which we use for this project. The application of the Poisson Binomial distribution to the generalized linear model setting has been discussed by Chen and Liu [4], who propose it for hypothesis testing on the parameter vector for a logistic regression model. But we find no references in the literature to our appraoch: building generalized linear models for Poisson Binomial-distributed sums.

There is a richer body of literature on our chosen problem: modeling individual voter preferences in elections. In applied papers, researchers make use of a variety of tools to obtain labeled datasets: voter surveys [5], exit polls [2], or modeling whether an individual voted (which is public record), rather than her choice of candidate [11]. Armed with datasets in which both features and outcomes are

4. Dataset

4.1. Overview

We have collected two relevant datasets for our project that we combined together.

Our dataset of **Pennsylvania precinct-level election results** contains the total number of votes received by each candidate in the 2016 presidential election. We obtained this dataset from OpenElections[9].

The second is the **Pennsylvania voter dataset** which we obtained directly from the Pennsylvania Department of State[8]. This dataset contains a row for every registered voter in Pennsylvania as well as their party registration, limited demographic information (age, gender), and voting participation over a set of recent primaries and general elections.

For the milestone, we focused on four of Pennsylvania's 67 counties: Adams, Allegheny, Bedford, and Chester. These counties are diverse – Allegheny and Chester were won by Clinton while Adams and Bedford were won by Trump. Together they represent 1,639 precincts and about 950,000 total votes for Clinton and Trump.

4.2. Dataset Preparation and Validation

Because these files were sourced from two different datasets, we ran into some challenges in cleanly mapping them with each other. We had to match precincts between the two files based on name instead of some sort of unique code or identifier. After our initial checks these worked fine for the four counties we selected for the milestone, and one of our next steps will be to investigate this further and get a mapping across all precincts and counties in Pennsylvania to have more usable data.

We only model based on the vote counts for Hillary Clinton and Donald Trump because the vast majority of votes cast were for one of them. Our datasets are not perfectly accurate because some voters voted for candidates besides Clinton and Trump and also because voters can fill out a ballot but not select a presidential candidate, voting only in other races (e.g. local ones) instead. Upon examining our dataset we noticed that the sum of Clinton/Trump votes in precincts is not equal to the total number of votes cast in the precinct, but is within 10% in most cases. To address these data issues, we take the percentage of Clinton/Trump votes cast in a precinct went to Clinton and multiply it by the total number of Clinton/Trump votes in the precinct, and repeat this for Trump. This number is our estimate of how many voters in a precinct would have went for Clinton/Trump if these two candidates were the only two options.

5. Methods

5.1. Poisson Binomial GLM form

We use a Generalized Linear Model based on the Poisson binomial distribution. We model an individual i voting for Clinton as a Bernoulli random variable, so $p_i = \sigma(\theta^T X_i)$, where $\sigma(.)$ denotes the sigmoid function, θ is a set of parameters to fit, and X_i are known covariates from the Pennsylvania voter dataset. Note that the probability of an individual voting not voting for Clinton (i.e. voting for Trump) is $1 - p_i$. We assume that these Bernoulli random variables are independent but not necessarily identically distributed, since we expect that different voters would have different probabilities of voting for Clinton.

Combining this representation of a voter with the fact that in a given precinct we know the total number of votes for Clinton and Trump, the total number of Clinton voters in each precinct will follow a Poisson binomial distribution, which is the probability distribution of a sum of independent but not necessarily identically distributed Bernoulli random variables[4]. For a precinct k with D votes for Clinton out of T total votes, the likelihood is given by:

$$\ell_k(\theta) = \sum_{A \in F_k} \prod_{i \in A} p_i \prod_{j \in A^c} (1 - p_j)$$

where F_k is the set of all configurations of T votes in which a total of D votes were cast for Clinton; A is the set of voters who voted for Clinton under that configuration, and A^c is the set of voters who voted for Trump under that configuration. The likelihood of the precinct-level results given parameters θ can be calculated by multiplying the likelihoods from every precinct together:

$$\ell(\theta) = \prod_k \ell_k(\theta)$$

In order to determine the optimal parameters θ, we need to maximize this likelihood over the Pennsylvania precincts. Note that the Poisson binomial likelihood involves sums over all possible configurations of votes – e.g. if Clinton received 200 out of 500 total votes in a precinct, then the likelihood involves a sum over $\binom{500}{200}$ configurations. Although we can directly estimate the likelihood using the `poibin` package, calculating the gradient is computationally infeasible.

5.2. Calculating the Gradient

To address this problem, we make use of the Lyapunov CLT to observe that the asymptotic distribution of d_k, the number of votes for Clinton in precinct k, is given by:

$$d_k \xrightarrow{d} N\left(\sum_i p_{k,i}, \sum_i p_{k,i}(1-p_{k,i})\right)$$

where $p_{k,i}$ is the i^{th} entry of p_k. This result is proven in the appendix. It allows us to *estimate* the likelihood with a much simpler function of θ. In this case, the contribution of precinct k to the overall log-likelihood is approximately:

$$\ell_k = -log\left(\phi_k\right) + \frac{1}{\phi_k^2}\left(d_k - \mu_k\right)^2$$

where irrelevant constants have been dropped, $\mu_k = \sum_i p_{k,i}(1-p_{k,i})$, $\phi_k^2 = \sum_i p_{k,i}(1-p_{k,i})$, and $p_{k,i} = \sigma(\theta^T x_{k,i})$. This yields a gradient of the form:

$$\nabla_\theta \ell_k = \frac{1}{2}\left(\frac{(d_k-\mu_k)^2}{\phi^4} - \frac{1}{\phi^2}\right)\left(\sum_i (2p_{k,i}-1)p_{k,i}^2 x_{k,i}\right) + \frac{1}{\phi^2}(d_k-\mu_k)\left(\sum_i p_{k,i}(1-p_{k,i})x_{k,i}\right)$$

5.3. Neural Net

As one potential improvement to our modeling, we considered using a neural net – rather than a simple logistic regression – to relate individual-level covariates to the probability of voting for Clinton. In particular, our new model for $p_{k,i}$, the Clinton-voting probability for person i in precinct k, is given by:

$$\begin{aligned} h_{k,i} &= \sigma\left(W_1 x_{k,i} + b_1\right) \\ p_{k,i} &= \sigma\left(W_2 h_{k,i} + b_2\right) \end{aligned}$$

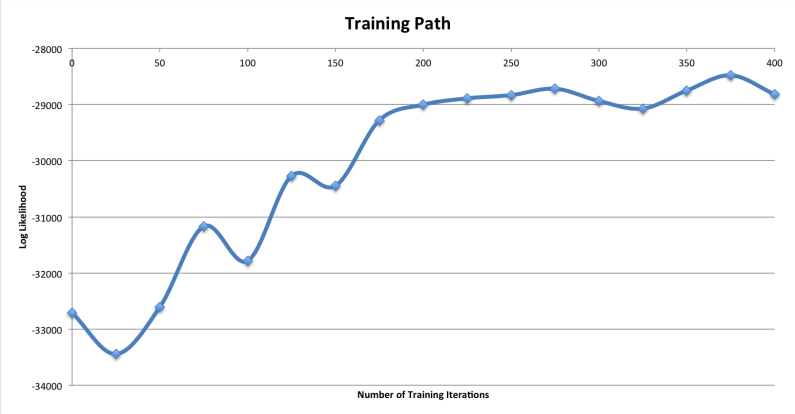
where $x_{k,i}$ are the covariates for the individual. Denoting $\mu_k = \sum_{i=1}^{n_k} p_{k,i}$ and $\sigma_k^2 = \sum_{i=1}^{n_k} p_{k,i}(1-p_{k,i})$ where n_k is the number of votes cast in precinct k, we see:

$$\begin{aligned} \frac{\partial \ell_k}{\partial p_{k,j}} &= -\frac{1-2p_{k,j}}{2\sigma_k^2} + \frac{1-2p_{k,j}}{2\sigma_k^4}(d_k-\mu_k)^2 + \frac{1}{\sigma_k^2}\left(d_k-\mu_k\right) \\ \frac{\ell_k}{\partial b_2} &= \sum_{j=1}^{n_k} \frac{\partial \ell_k}{\partial p_{k,j}} \cdot p_{k,j}(1-p_{k,j}) \\ \frac{\ell_k}{\partial W_2} &= \{h_{k,j}\}_j^T \left\{\frac{\partial \ell_k}{\partial p_{k,j}} \cdot p_{k,j}(1-p_{k,j})\right\}_j \\ \frac{\ell_k}{\partial b_1} &= \sum_{j=1}^{n_k} \left\{\frac{\partial \ell_k}{\partial p_{k,j}} \cdot p_{k,j}(1-p_{k,j})\right\}_j W_2^T \circ \{h_{k,j}(1-h_{k,j})\}_j \\ \frac{\ell_k}{\partial W_1} &= \{x_{k,j}\}_j^T \left\{\frac{\partial \ell_k}{\partial p_{k,j}} \cdot p_{k,j}(1-p_{k,j})\right\}_j W_2^T \circ \{h_{k,j}(1-h_{k,j})\}_j \end{aligned}$$

where $\{x_j\}_j$ denotes a column vector consisting of the entries x_j and $\circ$ denotes a Hadamard product. We implement these gradients and train our new neural net

6. Results

Using stochastic gradient ascent, we fit a simple model to the four counties where we predict the probability to vote for Clinton given county, party registration, primary participation, gender, and age. Computing the log-likelihood every 25 iterations, we see a modest improvement over the first 400 iterations (after which the log-likelihood plateaus):



Below we present the coefficients coming out of the model, noting that positive values indicate that a candidate is more likely to vote for Clinton over Trump:

Coefficient	Fitted Value
is Chester resident?	0.61
is Adams resident?	-0.42
is Bedford resident?	-0.78
is Allgeheny resident?	0.39
is registered Democrat?	0.30
is registered Republican?	-0.52

Coefficient	Fitted Value
voted in Democratic primary?	0.18
voted in Republican primary?	-0.30
is Female?	0.03
is Male?	-0.05
Age	-0.26

These results are in line with what we would expect. For example, older voters are more likely to vote for Trump and registered Democrats are more likely to vote for Clinton.

7. Next Steps

Our initial results are encouraging and we have a few key next steps as we proceed with our project. Our main next step is to run our model with more data, using as much of Pennsylvania as we can use. We would also like to fit more complex models with more predictors, possibly incorporating data from sources beyond the Pennsylvania voter file itself such as demographics if we can obtain them. As we obtain more data and fit more complex models, it will be important to make sure that we are converging efficiently. We are currently performing vanilla stochastic gradient descent, and we will investigate other optimization techniques such as Adam.

Once we are able to train on more data, we will split our data into training and test sets instead of training on all of it. Our plan is to train our model on 70% of the precincts in Pennsylvania and then test on the remaining 30% to gauge model accuracy, using the total number of Clinton and Trump votes per precinct as the outcome variable. We are are particularly curious to see if our approach (modeling at the individual level) can yield better predictions than modeling at the precinct level by aggregated individual-level features. There are many approaches to modeling at the precinct level (e.g. linear regression, neural nets) that we can compare against. We will also see if we can compare our approach against existing voter models that use polling data.

Once we begin training our model on only a training set and evaluating it on a holdout, we will experiment with adding regularization to see if it improves performance. In addition to gradient descent, we would also like to explore alternative methods for maximizing the log likelihood such as numerical gradients and expectation maximization.

Appendix: Lyapunov CLT Proof

Define $d_k = \sum_{i=1}^{n} d_{k,i}$ to be the number of Democratic votes in precinct k, where $d_{k,i}$ is an indicator variable denoting whether person i in precinct k voted for Clinton. d_k follows a Poisson binomial distribution with success probabilities

$p_k = (p_{k,1}, \ldots, p_{k,n})$. Define $s_k^2 = \sum_{i=1}^n p_{k,i}(1 - p_{k,i})$. We check the Lyapunov CLT [1] condition for the fourth moment:

$$\lim_{n\to\infty} \frac{1}{s_k^4} \sum_{i=1}^n E\left((d_{k,i} - p_{k,i})^4\right) = \lim_{n\to\infty} \frac{\sum_{i=1}^n p_{k,i}(1 - p_{k,i})\left(3p_{k,i}^2 - 3p_{k,i} + 1\right)}{\left(\sum_{i=1}^n p_{k,i}\left(1 - p_{k,i}\right)\right)^2} \stackrel{?}{=} 0$$

Observe that $3p_{k,i}^2 - 3p_{k,i} + 1 \in (0, 1)$ if $p_{k,i} \in (0, 1)$. Hence, the numerator is strictly less than $\sum_{i=1}^n p_{k,i}(1 - p_{k,i})$. Hence, if we can guarantee the numerator grows without bound, then this limit is 0 and the Lyapunov CLT applies. We can do so using a simple condition, like enforcing that there is some $\epsilon > 0$ such that $\epsilon < \bar{p}_i < 1 - \epsilon$ for all i (i.e. the mean probability of voting for Clinton in a precinct never falls below some low threshold ϵ or above some high threshold $1 - \epsilon$).

The Lypaunov CLT now tells us that:

$$\frac{d_k - \sum_{i=1}^n p_{k,i}}{s_k} \xrightarrow{d} N(0, 1)$$

giving us the desired asymptotic normality.

8. Contributions

Evan and Nitin both worked together to define the problem and scope it out as a Poisson GLM, and find the Pennsylvania state and OpenElections datasets. Evan wrote the majority of code for the progress made so far for the milestone and derived the CLT proof, while Nitin wrote up the milestone report as well as additional sections that will be useful for the final project report and focused on establishing better data mappings between our two datasets.

References

[1] P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley, 1995.

[2] T. M. Carsey. The contextual effects of race on white voter behavior: The 1989 new york city mayoral election. *The Journal of Politics*, 57(1):221–228, 1995.

[3] L. H. Y. Chen. On the convergence of poisson binomial to poisson distributions. *Ann. Probab.*, 2(1):178–180, 02 1974.

[4] S. X. Chen and J. S. Liu. Statistical applications of the poisson-binomial and conditional bernoulli distributions. *Statistica Sinica*, 7(4):875–892, 1997.

[5] J. K. Dubrow. Choosing among discrete choice models for voting behavior. 2007.

[6] W. Ehm. Binomial approximation to the poisson binomial distribution. *Statistics & Probability Letters*, 11(1):7 – 16, 1991.

[7] Y. Hong. On computing the distribution function for the poisson binomial distribution. *Computational Statistics & Data Analysis*, 59(Supplement C):41 – 51, 2013.

[8] C. of Pennsylvania. Pennsylvania full voter export. `https://www.pavoterservices.pa.gov/Pages/PurchasePAFULLVoterExport.aspx`, 2017.

[9] OpenElections. Openelections data for pennsylvania. `https://github.com/openelections/openelections-data-pa`, 2017.

[10] B. Roos. Asymptotics and sharp bounds in the poisson approximation to the poisson-binomial distribution. *Bernoulli*, 5(6):1021–1034, 12 1999.

[11] T. Rusch, I. Lee, K. Hornik, W. Jank, and A. Zeileis. Influencing elections with statistics: Targeting voters with logistic regression trees. *The Annals of Applied Statistics*, pages 1612–1639, 2013.

[12] M. J. Straka. Poisson binomial probability distribution for python. `https://github.com/tsakim/poibin`, 2016.

[13] N. Y. Times. Pennsylvania presidential race results. `https://www.nytimes.com/elections/results/pennsylvania-president-clinton-trump`, 2017.