Using Poisson Binomial GLMs to Reveal Voter Preferences

Evan Rosenman
Stanford University
rosenman@stanford.edu

Nitin Viswanathan
Stanford University
nviswana@stanford.edu

December 15, 2017

1. Introduction

For American political parties to compete effectively, it is vital to obtain accurate models of individual voter preferences. Using such models, parties can identify swing voters and focus outreach efforts ahead of elections.

In the U.S., vote tallies for every candidate in a given race are 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. However, state-level polling data can be inaccurate, as it was in the 2016 presidential election. To address this problem, we develop individual voter models based off of publicly available precinct-level voting data.

We develop a new type of generalized linear model (GLM), the Poisson binomial generalized linear model, for this task. To our knowledge this GLM formulation has not previously been implemented in the literature.

We fit this model to the results from the 2016 presidential election in Pennsylvania, a key swing state that, in an upset, favored Donald Trump over Hillary Clinton by a margin of 0.72% [25]. The model is fit by minimizing the negative log-likelihood of the Poisson binomial, allowing us to discover associations between individual voting preferences and key covariates. We then verify the accuracy of our model on a holdout set.

Our paper makes two key contributions:

- We develop the math necessary to use and train Poisson binomial GLMs
- We apply Poisson binomial GLMs to the specific task of revealing voter preferences

2. 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 [7, 21, 3]. Prior research [12] has identified a closed-form expression for the CDF, which relies on the discrete Fourier Transform. This technique is leveraged in the `poibin` package [23], 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.

There is a richer body of literature on modeling voter preferences. Most of this research separates into two primary methodologies. In the first, researchers are interested in the relationship between voter characteristics and ballot preferences. To obtain labeled datasets, researchers use voter surveys [6] and exit polls [2]. They then fit models via simple GLMs, like the multinomial probit or multinomial logit model, which "represent voter choice as a decision among unordered alternatives (parties, candidates) as a function of chooser (voter) . . . attributes" [5].

In the second methodology, researchers are interested in the relationship between candidate characteristics or voting methods and vote outcomes. In these studies [15, 10, 11], researchers frequently use aggregate vote totals from precincts, counties, or states. Relationships are uncovered by linear regression techniques, often using some random effects [13] and modified to include constraints. Closely related to this literature is the approach of popular election prognosticators like FiveThirtyEight [22] to model election outcomes using a mix of polling and demographic data.

Lastly, the problem of "learning individual-level associations from aggregate data" [9] has precedent in modern machine learning literature [18, 14, 24, 19]. In this work, individual-level probabilities are estimated, frequently by applying kernels to the covariates and then using a probit or logit link. The fitting methods, however, rely on Bayesian techniques rather than asymptotic normality. In the political setting, Flaxman et al. used individual-level covariates to analyze the 2012 [9] and 2016 [8] elections. Their approach does not estimate person-level probabilities but rather uses kernel methods to relate individual features to aggregate statistics. Thus, we believe our fitting procedure via the normal approximation to the Poisson binomial distribution is novel.

3. Dataset

3.1. Overview

We combined two disparate datasets for our project. Our dataset of **Pennsylvania precinct-level election results** contains the total number of votes received by each candidate by precinct in the 2016 presidential election. We obtained this dataset from OpenElections [17]. Our other

Table 1. Pennsylvania precinct-level election results

County Name	Precinct Name	Candidate Name	Number of votes
MONTGOMERY	ABINGTON W1 D1	HILLARY CLINTON	603
MONTGOMERY	ABINGTON W1 D1	DONALD TRUMP	388
...	...	...	...

Table 2. Pennsylvania voter file

County Name	Precinct Name	Voter Name	Gender	Age	Other Attributes
MONTGOMERY	ABINGTON 1-1	Jane Doe	Female	27	...
...	...	...	...	...	...

dataset is the **Pennsylvania voter file** which we obtained directly from the Pennsylvania Department of State [16]. 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.

3.2. Dataset Preparation and Validation

Because these files were sourced from two different datasets, we ran into challenges in cleanly mapping them with each other. In particular, there is no shared precinct identifier across the files – precinct names often did not match between the files as in the above examples, and there was no other common precinct identifier. As a result, we had to review all 9,000+ precincts manually to determine the best way to match them between the files. If either the voter file or precinct-level results were corrupted or did not match which each other, we removed the entire county from our dataset.

After our data cleaning, filtering, and mapping, we ended up with a dataset of 48 of Pennsylvania's 67 counties. Together, these counties represent 6,837 precincts and about 4.07 million total votes for Clinton and Trump. This corresponds to about 66% of voters in Pennsylvania. The sample had a small pro-Clinton bias, with 52.1% of the voters who cast a major-party ballot supporting Clinton in our sample, versus 49.6% statewide.

We only model based on the vote counts for Hillary Clinton and Donald Trump because about 96% of votes went for one of them. We examined this further and saw that the sum of Clinton/Trump votes in some precincts is not equal to the total number of votes cast, but is within 10% in the vast majority of cases. To address this, we take the percentage of Clinton/Trump votes cast in a precinct that went to Clinton and multiply it by the total number of votes in the precinct, and repeat this for Trump. This is our estimate of how many voters in a precinct would have chosen Clinton or Trump if these two candidates were the only two options.

4. Methods

4.1. Poisson Binomial GLM

We use a generalized linear model based on the Poisson binomial distribution. We model an individual i in precinct k voting for Clinton as a Bernoulli random variable with success probability $p_{k,i} = \sigma(\theta^T x_{k,i})$, where $\sigma(\cdot)$ denotes the sigmoid function, θ is parameter vector to fit, and $x_{k,i}$ are known covariates for voter i from the Pennsylvania voter file. The probability of voting for Trump is $1 - p_{k,i}$. 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_k votes for Clinton out of T_k total votes, the log likelihood is given by:

$$\ell_k(\theta) = log \left(\sum_{A \in F_k} \prod_{i \in A} p_{k,i} \prod_{j \in A^c} (1 - p_{k,j}) \right)$$

where F_k is the set of all configurations of T_k votes in which a total of D_k 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 log likelihood of the precinct-level results given parameters θ can be calculated by adding the likelihoods from every precinct together:

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

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.

4.2. Calculating the Gradient

To address this problem, we make use of the Lyapunov CLT [26] to observe that the asymptotic distribution of D_k is given by:

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

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

$$\ell_k(\theta) \approx -log\left(\phi_k\right) + \frac{1}{\phi_k^2}\left(D_k - \mu_k\right)^2$$

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

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

4.3. Neural Network

As one potential improvement, we also looked at using a neural network rather than a logistic regression to relate individual-level features 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:

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

where W_1, W_2 are weight matrices and b_1, b_2 bias vectors. We can compute the gradients:

$$\frac{\partial \ell_k}{\partial p_{k,j}} = -\frac{1-2p_{k,j}}{2\phi_k^2} + \frac{1-2p_{k,j}}{2\phi_k^4}(d_k - \mu_k)^2 + \frac{1}{\phi_k^2}\left(d_k - \mu_k\right)$$

$$\frac{\partial \ell_k}{\partial b_2} = \sum_{j=1}^{n_k} \frac{\partial \ell_k}{\partial p_{k,j}} p_{k,j}(1-p_{k,j})$$

$$\frac{\partial \ell_k}{\partial W_2} = \{h_{k,j}\}_j^T \left\{\frac{\partial \ell_k}{\partial p_{k,j}} p_{k,j}(1-p_{k,j})\right\}_j$$

$$\frac{\partial \ell_k}{\partial b_1} = \sum_{j=1}^{n_k} \left\{\frac{\partial \ell_k}{\partial p_{k,j}} p_{k,j}(1-p_{k,j})\right\}_j W_2^T \circ \{h_{k,j}(1-h_{k,j})\}_j$$

$$\frac{\partial \ell_k}{\partial W_1} = \{x_{k,j}\}_j^T \left\{\frac{\partial \ell_k}{\partial p_{k,j}} \cdot p_{k,j}(1-p_{k,j})\right\}_j W_2^T \circ \{h_{k,j}(1-h_{k,j})\}_j$$

where $\{x_j\}_j$ denotes a column vector consisting of the entries x_j and $\circ$ denotes a Hadamard product.

For our experiments, we construct a neural network with 1 hidden layer with 10 neurons. We experimented with other sizes for the hidden layer but saw little change in performance.

4.4. Optimization Methodology

Our goal is to obtain an accurate representation of the probability of any given voter to vote for Clinton. We train our model using gradient descent where the log-likelihood and gradients for our runs are as defined in sections 4.2 and 4.3. For each training epoch, we train our model on every precinct in the training set, looping through them in the same order every time and updating parameter values after each precinct. We explored using batch and stochastic gradient descent, but both of these resulted in slower convergence. We used a learning rate of 0.0001 with annealing at a $n^{-1/2}$ decay.

In the logistic regression case, we initialize our coefficients to zero (yielding a 50% Clinton-voting probability for every voter). For the neural net, we initialize W_1, W_2 to small random uniform values in $[-0.1, 0.1]$. We saw no overfit issues with the logistic regression model, but a small tendency to overfit in the neural network. We experimented with using an L_2 penalty for the neural net but were not able to get a performance improvement on the test set.

One specific issue we ran into when running gradient descent was extremely large or small gradients (that often returned as NaN due to underflow issues). We handled small gradients by ignoring them, as they would not have changed the likelihood at all, and we handled large gradients by capping them.

We evaluated the fit of our model both by ensuring that the loss (i.e. negative log-likelihood) decreased over time and also by looking at training vs. test R^2 values. Decreasing loss over training epochs indicates that our model is improving, and we look for it to eventually stabilize which indicates that the parameter values have approximately converged. When training our model, we consistently reached convergence within 20 epochs.

5. Results & Analysis

5.1. Evaluating model accuracy

First, we wanted to show that our models were accurate in predicting voter preferences in aggregate. We did this by training our model on 70% of the counties in our dataset and evaluating it on the remaining 30% in different ways.

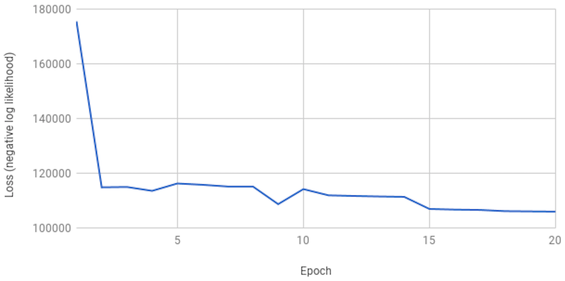

Figure 1. Training loss for logistic regression voter model

We can see in Figure 1 that our training loss decreases over time and converges after around 10 epochs. We saw similar loss decrease and rates of convergence for our neural network model as well.

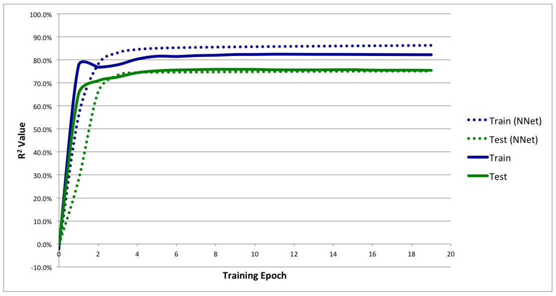

Figure 2. Training and Test R^2 for logistic regression voter model

Figure 2 shows model R^2 when predicting vote proportions within each *precinct*, for both the training and holdout sets. Results are weighted by the vote count in the precinct.

The training paths are similar whether we use logistic regression or a neural network to model individual voter behavior. The final training set R^2 for the logistic regression was 82.2% for training and 75.4% for test. For the neural network, the respective values were 86.3% and 75.1%. Training set accuracy is higher in the neural network model, but test set accuracy is slightly lower. The neural network's additional complexity does not help us better model the data. We also tried adding L_2 regularization by setting $\lambda = 0.7$ as well as trying some other values, but we saw no improvement to test set R^2. As a result, we ran our subsequent experiments only with the logistic regression model for individual voter behavior.

5.2. Weak labeling results

After we were convinced that our model performed well at a higher level, we then wanted to show model accuracy at the individual voter level. However, due to the nature of voting data and secret ballot, we do not actually have any labeled data that we can use to validate our model. To get around this, we derived and used weak labels as an approximate way to gauge our model's performance. Weak labels have been studied in prior literature and are used in supervised learning problems where either no labeled data is available or the available data is insufficient [20].

We apply two different forms of weak labels to gauge model performance at the individual voter level. First, we evaluate our model only against landslide precincts in the test set, where we define a landslide precinct to be one where 90% or more of voters supported the same candidate. We evaluate our model against voters in these precincts and expected to see the model predicting very high probabilities of voting for Clinton for voters in precincts that went to Clinton and predicting very low probabilities of voting for Clinton for voters in precincts that went to Trump.

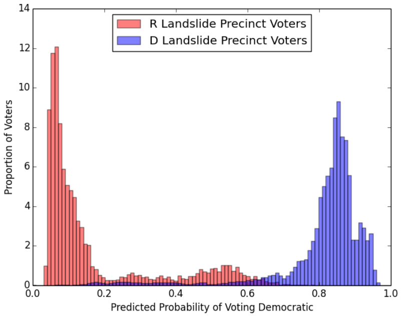

Figure 3. Predictions for voters in landslide precincts

Figure 3 has the expected bimodal distribution – the voters are either heavily in favor of or heavily against Clinton.

In our second weak labeling validation, we assume that all primary voters supported their party's eventual nominee in the general election. This is a reasonable assumption, as primary voters tend to have strong party loyalty. To evaluate, we build our model on the training counties and removed the feature indicating whether or not voters voted in their respective primaries. We then evaluate our model only on test set voters who voted in a primary.

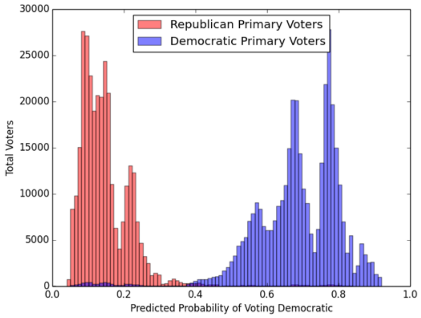

Figure 4. Predictions for primary election voters

Figure 4 has the expected bimodal distribution, indicating the model has learned that primary voters are indeed more likely to vote along party lines in the general election.

5.3. Final model trained on all data

After confirming our model's predictive validity, we trained it on our entire dataset to build our final voter model. Coefficients are available in the below table. Positive values correspond to a higher probability of voting for Clinton.

Model Term	Value
Apartment Dweller	1.231
Registered Democrat	1.065
County College Educated %	0.306
Voted in the Democratic Primary	0.287
County Population Density	0.218
Voted Absentee 2012	0.152
is Female	0.089
Voted Absentee 2014	0.083
County Latino %	0.045
County Black %	0.032
Voted In Person 2014	-0.077
County Income	-0.109
County White %	-0.119
Voted In Person 2012	-0.178
Voted in the Republican Primary	-0.519
Age	-0.542
Registered Republican	-1.339

Table 3. Model coefficients. Positive values correspond to a higher probability of voting for Clinton

These results are in line with our expectations. Democrats, younger voters, urban voters, and women are more likely to vote for Clinton while the opposite is true of Trump. These results increase confidence that we have developed an explanatory model for voter preference.

Conclusion and Future Work

We have combined aggregated precinct-level election data with information about individual voters to build a model predicting how individuals will vote. We implemented both logistic regression and neural network models to relate individual covariates to voting probabilities. We then trained these models via gradient descent, making use of a normal approximation to the Poisson binomial, which we derived. Our model's accuracy was validated using both aggregate vote totals and weak labels. After validating model accuracy, we retrain the model on the entire dataset to obtain our final model revealing individual voter preferences for Pennsylvania during the 2016 presidential elections. We believe this model would be a valuable resource for political organizations in voter targeting.

Our work contributes to the literature on learning individual associations from aggregate data, and is one of only a small number of such papers to work directly on political data. Our fitting method appears to be entirely novel, as we find no references in the literature to leveraging the Poisson binomial distribution in this way.

For future work, we'd like to further explore the idea of using neural networks to compute individual voter probabilities. With further exploration and tuning of hyperparameters, we could likely obtain a neural network model that outperforms the logistic regression model. We could also look at other regularization schemes such as dropout to reduce the gap between the training and test performance. In addition to investigating neural networks further we could also look at other link functions, such as the probit, to model individual voter behavior from covariates.

We would also like to revisit the data for counties that we were not able to use for our initial analysis so we can incorporate them into the analysis as well, and develop data pipelines that would allow us to efficiently repeat this analysis on other states and elections throughout the U.S.

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:

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

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 $\sum_{i=1}^n p_{k,i}(1 - p_{k,i})$. Thus, 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 infinitesimal threshold ϵ or above $1 - \epsilon$).

The Lypaunov CLT now tells us that:

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

6. 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 optimization algorithm and derived the CLT proof, while Nitin wrote the majority of the report and cleaned the data to allow us to use as many counties/precincts as possible. Both Evan and Nitin reviewed each other's work in addition to the parts they led.

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. Dow and J. W. Endersby. Multinomial probit and multinomial logit: a comparison of choice models for voting research. *Electoral Studies*, 23(1):107 – 122, 2004.

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

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

[8] S. Flaxman, D. Sutherland, Y.-X. Wang, and Y. W. Teh. Understanding the 2016 us presidential election using ecological inference and distribution regression with census microdata. *arXiv preprint arXiv:1611.03787*, 2016.

[9] S. R. Flaxman, Y.-X. Wang, and A. J. Smola. Who supported obama in 2012?: Ecological inference through distribution regression. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 289–298. ACM, 2015.

[10] L. Frisina, M. C. Herron, J. Honaker, and J. B. Lewis. Ballot formats, touchscreens, and undervotes: A study of the 2006 midterm elections in florida. *Election Law Journal*, 7(1):25–47, 2008.

[11] M. C. Herron and D. A. Smith. Race, party, and the consequences of restricting early voting in florida in the 2012 general election. *Political Research Quarterly*, 67(3):646–665, 2014.

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

[13] J. N. Katz and G. King. A statistical model for multiparty electoral data. *The American Political Science Review*, 93(1):15–32, 1999.

[14] H. Kuck and N. de Freitas. Learning about individuals from group statistics. *arXiv preprint arXiv:1207.1393*, 2012.

[15] J. M. Miller and J. A. Krosnick. The impact of candidate name order on election outcomes. *Public Opinion Quarterly*, pages 291–330, 1998.

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

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

[18] G. Patrini, R. Nock, P. Rivera, and T. Caetano. (almost) no label no cry. In *Advances in Neural Information Processing Systems*, pages 190–198, 2014.

[19] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10(Oct):2349–2374, 2009.

[20] A. Ratner, S. Bach, P. Varma, and C. R. Weak supervision: The new programming paradigm for machine learning, Jul 2017.

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

[22] N. Silver. A users guide to fivethirtyeights 2016 general election forecast. `https://fivethirtyeight.com/features/a-users-guide-to-fivethirtyeights-2016-general-election-forecast/`, 2016.

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

[24] T. Sun, D. Sheldon, and A. Kumar. Message passing for collective graphical models. 2015.

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

[26] E. W. Weisstein. Lyapunov condition. `http://mathworld.wolfram.com/LyapunovCondition.html`.