Predicting Election Outcomes

Mohammad Assem Abdalqader Mahmoud

December 19, 2020

1 Introduction

1.1 Outline

This project report is mainly based on [Kos18], and complemented by [CF08]. We implement simulation methods for predicting the outcome of the US presidential election proposed in [Kos18, CF08]. The model results for the 2020 election, based on data from FiveThirtyEight (the presidential general election polls), will be produced. We will focus only on the frequentist approach, and will not present the Bayesian method.

1.2 Background

Is the U.S. president elected by a majority of the popular vote? Yes and No. They do need a majority of votes, not from the general U.S. citizens, but from electors known as the *Electoral College*. The Electoral college is a group of 538 electors (senators and representatives) each of them is appointed by a state to wisely vote for the president on behalf of its people. Such electors typically vote for the candidate who wins a majority in their state as they pledge to that. They may choose to be faithless and vote otherwise or abstain, but that is rare and never swung an election.

The number of *electoral votes* of a state is the number of the electors who represent that state in the Electoral College. Not all states have the same number of electoral votes. Some states are heavy in the sanse that they have many electoral votes, e.g. Florida has 29, and others have few, Alaska has only 3. The District of Columbia (DC) is treated like a state with 3 electors of its own. That being said, a candidate needs at least 270 ($> \frac{538}{2} = 269$) electoral votes for a win.

In summary, the U.S. presidential election is multistage: state-level, then electoral college, and a candidate wins the presidency if they collect at least 270 electoral votes.

1.3 Motivation and Related Work

The search for order in a seemingly chaotic political world is desirable for strategic planning. Analyses of popular vote trends do not address the complicated role played by the electoral college in this multistage election process.

Several research implemented regression models for state polls to forecast the outcome of the popular vote (e.g. [CW90, Cam92]). On a smaller level, for each state on its own, polls are used to predict the state's outcome (e.g. [Coh98, TJ99, SCJ06]). A multilevel logistic regression model to generate estimates of state-level votes was implemented in [PGB04, PGB17].

Here, following mainly Kostadinov's work in [Kos18], as well as the work by others in [CF08], we use state-level polls directly to model state-level outcomes using a frequentist as well as a Bayesian approach. After that, we pay attention to the role of the Electoral College which we simulate as coin (biased) tossing experiment.

1.4 Goals

To simply and precisely state our final goal here, we ask the reader to pretend that we are back in time, say we are now early October 2020, and we want to answer to the following question: **What is the probability of the event: Biden will win the election?**

Other goals, while working on aswering the question, include a successful implementation of the work in [Kos18, CF08], and manifesting the applicability of Monte Carlo simulation on the election prediction matter.

2 Methods

2.1 Bootstrapping

This is the frequentist approach. We use the bootstrap method ([J.69] then [B.82]) to resample from the observed poll data to set up a probability distribution representing all possible rersponses (Biden, Trump, Others) with the corresponding proportions (frequencies). We then use the distribution to simulate a large number of hypothetical poll samples for each state from which we infer, for Biden, the probability of him winning that state's electoral votes.

2.2 Simulating the toss of a biased coin

At this point, for each state k, we have a Biden-win probability p_k. We mimic the Electoral College role by tossing 55 coins to account for the nonstandard behavior of Maine and Nebraska as thir electoral votes are based on congressional-district-level electoral votes. Note that, in the data we have, one of Nebraska's districts is missing, otherwise we would be tossing 56 coins.

For simplicity, we refer to both districts and states as "states". A state k coin has a Biden head and a Traump tail, and is biased so that when tossed it lands Biden with probability p_k. Again, we proceed as frequentists and run this experiment a large number of times n. In each of the n scenarios we calculate the total number of the electoral votes for Biden. Then, we infer the probability of Biden winning the election by counting the number of scenarios in which the calculated number of Biden electoral votes is ≥ 270, then dividing that count by n.

In the paper [Kos18], the author also experiments simulating the electoral college with fair coins and compares the results with those using biased coins. We will do the same.

3 Results

3.1 Data-set Descriptions

FiveThirtyEight made it convenient accessing the pre-election state polling data. Although such data are inevitably flawed, they can still provide insights. Researchers have noted that this polling data may not be useful until early September after the two parties' national conventions [CW90, Cam96].

The data is collected by different pollsters which each has a grade according to FiveThirtyEight's quality grading system. For a signle state on a single day, the same pollster may include more than one reading for the single candidate.

In [Kos18], the author uses the data published by FiveThirtyEight for the 2016 election. They do not describe exactly how they use the polls' data. More precisely, they do not mention the poll dates and the pollsters they use. They only describe how to resample based on the data given by a single pollster on a specific day. In [CF08], for the analyses of the 2004 election, the authors gave a bit of details. They recorded poll updates at 12 different times beginning on October 12, 2004, and ending on November 2, the day before the election. They then predicted the state-win probabilities for Bush using three different data assimilation methods:

(1) Latest poll. Consider only the most recent poll for each state.

(2) Combined Polls. Combine all previous polls up to the present time and treat it as a single sample, weighting only by sample size.

(3) Weighted Polls. Like combined polls but adjust sample size according a weight function depending on the day the poll is taken. The weight function they used is $w(t) = 1 - \frac{t}{26}$, where t is the number of days since the poll was carried out.

The third method inspired us to use weighted polls, not only based on the date, but also based on the pollster's grade. However, the resulting sample size is too huge to bootstrap. The same applies for the second method, as it gives the same sample size. We tried to resolve this issue by picking the best pollster grade available for a given state, and used the sample combined from all the pollsters of that single grade. We still get a size which is too huge. A resolution that happened to be reasonable was considering a two week period before the last date available. This happened to be reasonable to manifest on a single state at a time, not with the 55 states. We finally decided use the period between 7/15/2020 and 10/7/2020 as during that period the sample sizes was manageable. Note that, those huge samples we obtained are rather an asset than a liability, however, they do not allow us to manifest the Bootstrap implementation here in this project.

3.2 Results at State-level (frequentist)

To estimate the probability of Biden winning the election, we begin by quantifying his popularity compared to Trump in each state. Recall that Maine and Nebraska are partitioned into several "substates" because these allow congressional districts to cast votes for different candidates. Maine has four electoral votes in the Electoral College, awards two electoral votes from the state at large, and one vote for each congressional district. Nebraska has five electoral votes in the Electoral College, two from the state at large, and one from each of its three congressional districts. In [Kos18], the author does not mention the within-state partitioning. In [CF08], they mention the partitioning without any following discussion as the districts' data were not readily available to them.

Let us start with an example similar to that given in [Kos18]. Consider a Florida Poll conducted by Siena College/New York (an A+ pollster) on October 13-26, 2020. The sample size was $N = 1424$ voters: 45.7% Biden, 44.17% Trump, and the rest is split across the other candiadates. The key idea behind the simulation is to use the poll's percentages to define a discrete distribution for the random variable S which has three possible outcomes: 0 (Biden), 1 (Trump), 2 (Other). In other words, we convert the state-level poll results into a trinomial vector consisting of the three probabilities.

For the Florida poll at hand, the distribution of S is specified by the probability vector $(0.457, 0.4417, (1 - 0.457 - 0.4417))$. We simulate the poll by resampling with replacement from $\{0, 1, 2\}$ a set of size N. This is a single scenario which we repeat thousands (10^4) of times then take average. The following pseudo code can explain things well:

N = 1424 (the sample size)
n = 4e4 (number of scenarios)
candidates = (0, 1, 2) (the 3 possible choices for every voter)
perc = (0.457, 0.4417, (1-0.457-0.4417)) (poll's percentages)
sim.polls = sample(candidates,size= (n,N) , replace=TRUE , prob=perc)
mean(rowSums(sim.polls == 0) > rowSums(sim.polls == 1))

As in [Kos18], we ran the simulation in R and obtained a 72.1278% chance that Biden would win Florida. We also ran the simulation in Python which is the language we use for the rest of this work.

We mentioned in the previous subsection that we will be assimilating the data using weighted polls. In order to not miss any states, we could not start our period on Oct 12, 2020, as in [CF08], but we had to start it July 15, 2020 (almost 1 month extra). For dates, we used the following weight function $w_d(date) = 1 - \frac{t}{112}$ where t is the time period in days between the date and the election day, and 112 is just 2 + 110 (which is the length of the whole period since July 15). For pollster grades, we ordered the 13 grades we have from 0 (best) to 12 (worst), and use the following weight function $w_g(grade) = 1 - \frac{v}{15}$ where v is the number given to the grade.

Such assimilation does result in a much larger sample size N since we include many pollsters and many days. For example, for Florida, the sample size becomes 684416 voters. This is a huge number compared to the number of scenarios, 10000, which we used before. It is not mentioned in [CF08] how many scenarios they used, but if we want to maintain the same ratio between sample size and number of scenarios as we did in the Florida example, then

we will require a 5×10^6 scenarios. This number of scenarios, using a very efficient algorithm, is enough to crash a Kaggle kernel. In fact, running only a 100 scenarios takes around 1.76s, and running a 1000 takes 18s (linear growth as expected). For more inofrmation, see the functions 'Assimilate 2' and 'Return-prob-Biden' and their outputs in the code1 attached.

We invented the following method to assimilate the data (function 'Assimilate 3' in the code1 attached). For each state, and each given date, we consider all the polls that have the highest quality grade available. We average on them to obtain the probability, and take the sample size to be the total. Using this, the sample size for Florida turned out to be 26698 which was also huge.

We invented another method which yields a smaller sample size by considering only few days ahead from the last date available for a given state. Here we still weighted the percentages using dates exactly as in the last method. If we consider 1 week ahead, the sample we get has 13803 voters. This is still a lot for bootstrapping. We finally decided to use only 3 days ahead which gives sample of size 9720 voters.

In this last sample, we have around 49.4% Biden, and 45.2% Trump. Now, a reasonable number of scenarios to run is 50000. This yields a probability 0.49896 that Biden (and 0.49668 for Trump) would win Florida. This simulation took 14.8s.

To work all the states, it was still a lot to even consider only the last day available, which is the first method mentioned in the previous subsection. The last day for Florida alone has 9264 voters (looking at only A+ pollsters). This is still a lot.

As we mentioned in the previous subsection, we managed to predict all the state by considering a period with less traffic, namely, between 7/15/2020 and 10/7/2020. This is presented in the attached code2.

3.3 Results for the Electoral College

Now that we have the Biden win probabilities for every state, we can simulate the Electoral College by tossing 55 biased coins as we mentioned before.

Suppose that for state number k, the probability obtained before that Biden would win is p_k. We mimic a single coin toss by the random variable $X_k = I(u < p_k)$ where I is the indicator functions and $u \sim U(0,1)$. A single toss experiment is equivalent to observing the value of X_k.

Based on that, the total number of electoral votes obtained by Biden is a the following random variable $eVotes$:

$$eVotes = \sum_{k=1}^{5} 5 v_k I(u_k < p_k)$$

where v_k is the number of electoral votes for state number k, and the $(u_k)_k$ are i.i.d. $U(0,1)$. If $eVotes$ is observed to be at least 270, then we have observed a Biden win.

Finally, we repeat this a large number of scenarios to obtain a Monte Carlo estimate. Python allows us to elegantly do so (check last cell code2).

4 Contributions Summary

(1) Considering a vriety of tricky assimilations of the data that were not presented or implemented in the papers.

(2) Building our own algorithms for the most part.

(3) Extending the results in the papers to consider the congressional districts in Maine and Nebraska.

(4) Producing the results for the 2020 election

References

[B.82] Effron B. The jackknife, the bootstrap, and other resampling plans. *SIAM Monograph 38, Society for Industrial and Applied Mathematics, Philadelphia*, 1982.

[Cam92] James E. Campbell. Forecasting the presidential vote in the states. *American Journal of Political Science*, 36(2):386–407, 1992.

[Cam96] James E. Campbell. Polls and votes: The trial-heat presidential election forecasting model, certainty, and political campaigns. *American Politics Quarterly*, 24(4):408–433, 1996.

[CF08] William F Christensen and Lindsay W Florence. Predicting presidential and other multistage election outcomes using state-level pre-election polls. *The American Statistician*, 62(1):1–10, 2008.

[Coh98] Jeffrey E. Cohen. State-level public opinion polls as predictors of presidential election results: The 1996 race. *American Politics Quarterly*, 26(2):139–159, 1998.

[CW90] James E. Campbell and Kenneth A. Wink. Trial-heat forecasts of the presidential vote. *American Politics Quarterly*, 18(3):251–269, 1990.

[J.69] Simon J. Basic research methods in social science. *Random House, New York*, 1969.

[Kos18] Boyan Kostadinov. Predicting the next US president by simulating the electoral college. *Journal of Humanistic Mathematics*, pages 64–93, jan 2018.

[PGB04] David K. Park, Andrew Gelman, and Joseph Bafumi. Bayesian Multilevel Estimation with Poststratification: State-Level Estimates from National Polls. *Political Analysis*, 12(4):375–385, 2004.

[PGB17] David K. Park, Andrew Gelman, and Joseph Bafumi. Bayesian multilevel estimation with poststratification: State-level estimates from national polls. *Political Analysis*, 12(4):375–385, 2017.

[SCJ06] Souren Soumbatiants, Henry W. Chappell, and Eric Johnson. Using state polls to forecast u.s. presidential election outcomes. *Public Choice*, 127(1/2):207–223, 2006.

[TJ99] Holbrook T.M. and Desart J.A. Using state polls to forecast u.s. presidential election outcomes in the american states. *International Journal of Forecasting*, 15:137–142, 1999.