1 Our model

In this section, we provide a quick review of our model. Let

$$Y_i = \frac{\text{Republican\%}}{\text{Republican\%} + \text{Democrat\%}} \tag{1}$$

be the outcome variable of interest, where i denote a district in a particular election. For Y being a 435-vector[1] representing elections in 2018, we form a linear probability prior

$$Y \sim \mathcal{N}(\mu_0, \Sigma_0) \tag{2}$$

where $\mu_0 = X\beta_0, \Sigma_0$ is estimated on the training data. To estimate β_0 and Σ_0 in (2), we use a cross-validated elastic net for β_0; this yields $\epsilon = Y_{\text{tr}} - \widehat{\mu}_0^{\text{tr}}$ on the training set. We consider two formats for Σ_0. With *diagonal* restriction, we simply let

$$\widehat{\Sigma_0}_{ii} = \frac{1}{n_i} \sum_{j:i\in\text{state}(j)} = \sum_j \frac{\mathbb{1}\left(i \in \text{state}(j)\right)}{\sum_{j'} \mathbb{1}\left(i \in \text{state}(j')\right)} \epsilon_j^2, \qquad n_i = |\{j : i \in \text{state}(j)\}| \tag{3}$$

be a state-smoothed estimate of variance on the training data. With *unrestricted* Σ_0, consider the vectors $\epsilon_{(i)}$ being ϵ entries corresponding to districts with the same label (e.g. AL–01) as i,[2] indexed by time. We compute[3]

$$\widehat{\Sigma_0}_{ij} = \kappa\widehat{\rho}_{ij}\left(\widehat{\Sigma_0}_{ii}\widehat{\Sigma_0}_{jj}\right)^{1/2}, \; i \neq j, \quad \widehat{\rho}_{ij} = \widehat{\text{Corr}}(\epsilon_{(i)}, \epsilon_{(j)}), \quad \kappa \in [0, 1] \tag{4}$$

where $\widehat{\text{Corr}}$ denotes the empirical correlation operator and κ is a shrinkage factor chosen so that the resulting estimate of Σ_0 is positive-definite. We assume that a poll outcome Z_j has a normal distribution conditional on Y: $Z_j \mid Y \sim \mathcal{N}(a_{Z_j}^T Y, \sigma_{Z_j}^2)$, where a_{Z_j} and $\sigma_{Z_j}^2$ are specified in our orginal report. This allows us to update the prior in (2) and arrive at a posterior

$$Y \mid Z \sim \mathcal{N}(\mu, \Sigma), \tag{5}$$

from which we generate our predictions by drawing from the posterior (5). The final predictions of the two models, diagonal and unconstrained, are plotted in Figure 1.[4]

This report is organized as follows. Section 2 gives an overview of our prediction quality compared to the FiveThirtyEight benchmark. Sections 3 to 5 discusses issues and methods in fitting μ_0, Σ_0, and the polling data, respectively. Section 7 concludes.

[1] In practice, we exclude the uncompetitive races from Y.

[2] Due to redistricting, the entries in $\epsilon_{(i)}$ could be completely unrelated to district i in 2018.

[3] The corresponding procedure in the original report is incorrect due to a computational error in the following expression—we did not raise the variance terms to 1/2-power in computing the off-diagonal covariance entries. As a result, the off-diagonal entries are small in our original report, and the resulting predictions are extremely similar. This is no longer the case once we correctly implemented (4)—see an email to Zhirui Hu on election day regarding this issue.

[4] [C] There was a comment regarding where the empty bars in the histograms in Figure 1 come from. We are calling the `seaborn.distplot` library function in Python. The data being plotted are integers, and so if the bin-size of the histogram is not integral, then we might see empty bins.

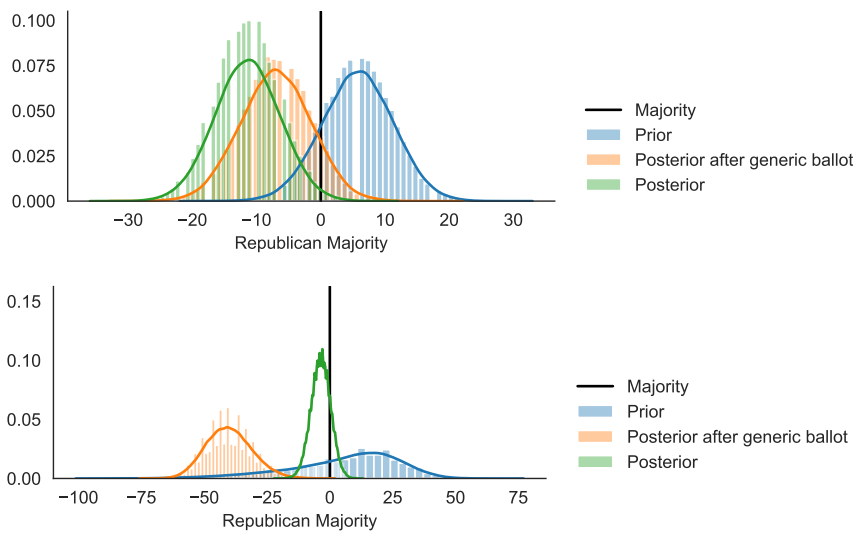

Figure 1: Model predictions. Top: diagonal. Bottom: unconstrained.

Table 1: Number of correctly called races for each model by winning party of each district (top pane), and number of expected seats won by Democrats compared to ground truth (bottom pane).

Winner	Total	Diagonal	Unconstrained	538
Democrat	240	216	211	226
Republican	195	190	187	193
Expected Democratic Seats	240	230	223	233

2 Overview of prediction quality

We plot a comparison of prediction quality between our prediction and that of FiveThirtyEight, broadly considered to be the state-of-the-art.[5] The R^2 by regressing actual vote share on predicted vote share and a constant for the diagonal model, unconstrained model, and FiveThirtyEight's model is 0.897, 0.893, and 0.967, respectively. Table 1 shows the number of correctly called races by model and winning party. Both Figure 2 and Table 1 show that the diagonal model performs better than the unconstrained model, and that both underperform relative to FiveThirtyEight's model. Moreover, Table 1 shows that all three models underestimated the Democrats' performance in the midterm elections, while the diagonal outperforms unconstrained, and both of our models underperform that of FiveThirtyEight's.

[5]We take the latest prediction for each district generated by FiveThirtyEight's house model (https://projects.fivethirtyeight.com/2018-midterm-election-forecast/house/, and transform the prediction into a statistic that corresponds to (1).

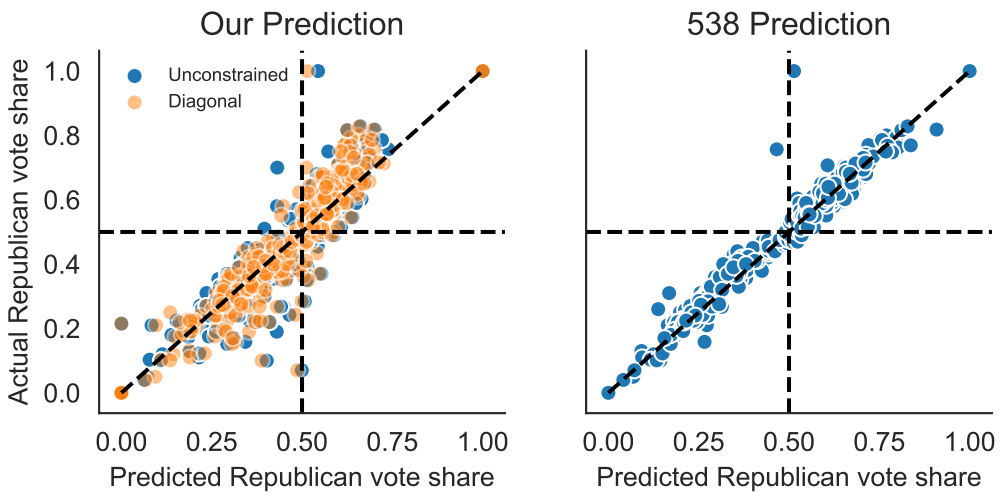

Figure 2: Quality of our prediction compared to that of FiveThirtyEight. The point that all predictions incurred large positive residuals is Alaska's district-at-large, where the Republican won a competitive race, yet the Republican's opponent is an Independent and not a Democrat. Thus (1) would define the response variable being 1, even though the race is fairly competitive.

It is clear from Table 1 and Figures 1 and 2 that the unconstrained model suffers from too little precision, as the correlation operator in (4) is extremely noisy, since the correlation is only taken over the four election years from 2010–2016. The unconstrained model was motivated by the fear that without modeling correlation of elections, the prediction model is going to be overly precise and would overlook systemic polling and modeling errors as was the case with the 2016 presidential election. However, it does seem that modeling correlation in the manner of (4) is not a good idea. From this point, we only consider the diagonal model.

The benefit of explicit probabilistic modeling in the manner that we have done is that we can evaluate the likelihood of the outcome that materialized. Consider the the distribution of the likelihood $\log L(\tilde{Y})$ of data $\tilde{Y} \sim \mathcal{N}(\mu, \Sigma)$ generated under our model-fitted posterior. The 0.1%-percentile of the likelihood is 630.35. With the general intuition of hypothesis testing, we reject the model if the observed data has extremely low likelihood under the model. Evaluating the data transformed via (1) suffers from extraordinarily low likelihood of a few observations, due to the presence of Independents (as in the caption of Figure 2). However, even with untransformed data, the data-likelihood is still exceptionally low compared to the purported data-generating distribution: The log-likelihood of transformed data (as in (1)) is 41.75, while the log-likelihood of untransformed Republican vote share is 442.25.

The reason of the low likelihood seem to be that the variance of the model being too low, as μ is fairly close to the materialized outcome, by Figure 2. We can consider a variance inflation parameter σ^2 which maximizes the data likelihood under model $\mathcal{N}(\mu, \sigma^2\Sigma)$. The fitted σ^2 is 4.262 for transformed data and 2.224 for untransformed data. Both estimates would strongly reject a null hypothesis of $\sigma^2 = 1$.

3 Fitting μ_0

3.1 Data and Preprocessing

In this section we consider the data and preprocessing issues in estimating μ_0 in (2). Much of the data that we collected are of limited reliability. For instance, the gender of the candidiate, upon manual inspection, was wrong in many cases. We used a Python `gender-guesser` library to account for this while marking gender neutral names as 0.5. Data reliability issues imply that (1) we could have improved our data cleaning process by looking up each candidate and filling in inconclusive entries, and (2) features such as presidential approval or education level may have suffered from inaccurate data reporting or exhibit high variance by being a point estimate as opposed to an aggregate over surveys from longer periods of time. **[TODO: This is not true, census data sample a lot of people; also, are we just saying that "given time we can clean data better?"]**

In Part I, we also discussed the effectiveness of Google Trends's relative popularity of search queries. However, there were several assumptions made that may have proved problematic. First, search frequency is a good proxy for interest, but it does not translate to support. We can imagine various scenarios such as an incumbent candidate not having as many search queries by virtue of already being well-known, or having a much higher search query due to a negative press release. As our model relies on a linear model with at most a second degree basis transform, tracing complex, non-linear relationships was limited without introducing overfitting. Second, for certain states (especially those with a low population), acquiring search frequency ratio of candidate names at a State-level granularity was impossible because a lack of absolute number of queries. We imputed these ratios with the ratios of a nation-wide search, which might introduce bias, as there are more people living in blue states than in red states. Last and not least, even for states where there are enough queries to return a state-level ratio, such as in Alaska (Figure 3), the resulting ratio alone may not be a discerning indicator for candidate preference.

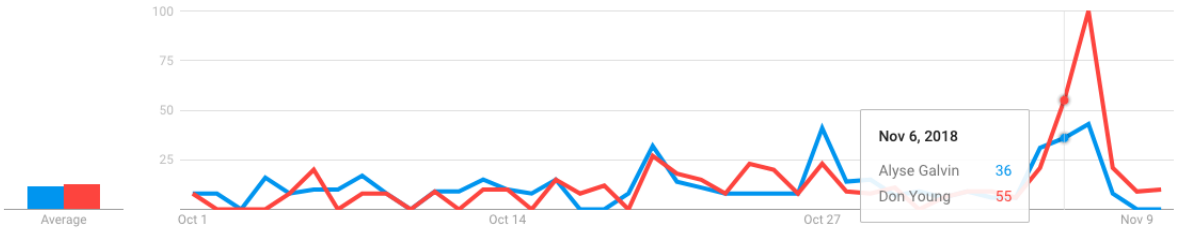

Figure 3: Google Trends search frequency ratio of Democratic vs. Republican candidate in Alaska at large.

A better approach to utilizing Google Trends data would be to have a list of key issues by party from institutional knowledge or from analyses of party platforms, and compare the interest of Republican flagship issues against Democratic ones.[6] For instance, Figure 4

[6]This approach would require some heavy topic/language modeling techniques in machine learning just

shows the Google Trends frequency results for four key issues: guns, defense, healthcare, and wellness. The first two are the flagship areas in which Alaska's Republican candidate emphasized, while the latter two are those prioritized by the Democratic candidate. The overall running average of these issues in Alaska suggests that there is ample room to augment our data by feeding in a more holistic picture of search query frequency ratios on topics that are brought forth by candidates. Don Young, the Republican, took Alaska's seat at the House.

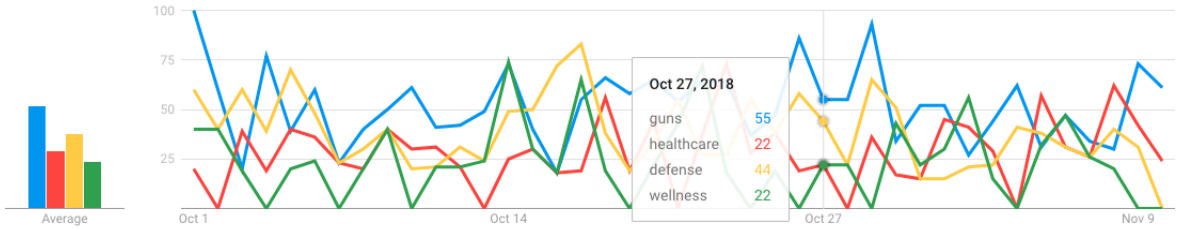

Figure 4: Google Trends search frequency ratio in Alaska on guns, healthcare, defense, and wellness. On average, guns are more frequently queried compared to healthcare, and defense more frequently than wellness.

3.2 Covariate and model selection in forming prior

We examine the effect of covariate selection by running the procedure while leaving one of the covariates out, somewhat mimicking a backward stepwise model selection procedure. Since the design matrix X included both linear and quadratic terms of the covariates, leaving one covariate out leaves out all its quadratic terms. We present the R^2 of the regression of the 2018 data on the predictions (as in Figure 2) in Table 2. We see that leaving out most covariates have little effect on the prediction quality, while racial makeup, incumbency, and educational background are particularly important for prediction. Moreover, for quite a few covariates, leaving them out actually *improves* fit, suggesting that the elastic net regularizer is not a panacea for overfitting—in particular, if the covariates have little predictive power, then in order for the regularizer to adequately control for overfitting, it must have a high level of shrinkage, which may result in underfitting, as the regularizer would discount variables that are highly predictive.

We also experiment with a some methods to improve model selection in Table 2, treating the data from 2018 as a validation set. In particular, the `lasso_select` entry in Table 2 represents a two-step procedure where a first-step LASSO regression is used to select covariates and a second-step elastic net is used to further control for overfitting and shrinkage. We see that this method marginally increases quality of fit. In `quadratic`, we simply leave out all quadratic terms in X, and rather discouragingly, this much sparser set of covariates perform better than both the `full` and the `lasso_select` models.

to collect the data.

Table 2: Effect on fit (among competitive races) of leaving one covariate (along with all higher-power terms that involve the covariate) out; we also show the performance of certain alternative prediction functions for μ_0. `full` means full model. `quadratic` means leaving out all quadratic terms. `lasso_select` means using a cross-validated LASSO to select covariates by discarding all covariates with zero fitted coefficent and running an elastic net on the rest of the covariates. `gradient_boost` is a gradient boosting regression tree with validation-guided early stopping. `logit` is a model where $\mu_0 = X\beta_0$ is replaced with $\mu_0 = \text{logit}^{-1}(X\beta_0)$—we fit an elastic net on logit-transformed training data to obtain β_0.

Variable left out / Model name	R^2	Variable left out / Model name	R^2
`minority_percentage`	0.7649	`same_as_pres_party_rep`	0.8215
`rep_incumbent`	0.8062	`dem_is_female`	0.8217
`percent_bachelor_or_above`	0.8086	`dem_percent`	0.8230
`gradient_boost`	0.8138	`rep_is_female`	0.8235
`pres_approval`	0.8177	`lasso_select`	0.8238
`log_med_income`	0.8182	`logit`	0.8259
`rep_to_tot_oct`	0.8183	`kernel_sigma`	0.8267
`rep_to_tot_nov`	0.8211	`dem_incumbent`	0.8272
`full`	0.8215	`quadratic`	0.8324
		`same_as_pres_party_dem`	0.8402

We also investigate whether alternative methods for fitting $\mu_0 = f_\beta(X)$, for some functional form f and parameters β, would have done better. We fit a gradient boosting regression tree (`gradient_boost`) on the same input space as the original model (`full`). We hold out a validation set and use an early stopping rule—stopping when the validation error fails to improve for a number of iterations.[7] `gradient_boost` does not appear to have better fit than the elastic net. We suspect that nonparametric methods like gradient boosting trees do not utilize the rich probabilistic information in the input data,[8] which results in a worse fit. This leads us to suspect that better specified probabilistic models should do better. We correct for the misspecification in (2) by fitting β on the logit-transformed space of the original data:

$$\text{logit}(Y_{\text{tr}}) \sim \mathcal{N}(X_{\text{tr}}\beta_0, \Sigma_{\text{tr}}),$$

mimicking the *logit-normal model* (Agresti, 2015, page 283)—so that the Normal distribution is properly specified on data that can take values in $\mathbb{R}$. However, modeling in the logit space does not lend well to the simple Bayesian updating in (5) without using expensive MCMC simulations, and for convenience's sake, in prediction, we modify (2) to $\mathcal{N}(\text{logit}^{-1}(X\beta_0), \Sigma_0)$, which is again a slight misspecification.[9] This model, `logit`, does appear to perform slightly

[7] We tried multiple hyperparameters for the early stopping; all of them failed to generate better fit than `full`.

[8] the probabilistic model (2) fits fairly well by inspecting a Q–Q plot, despite the misspecification; see original report for details

[9] Therefore, the `logit` model in Table 2 is not strictly-speaking a logit-normal model as in Agresti (2015)—rather an ad hoc alternative inspired by the logit-normal model in the literature.

better than the original model `full`, but the degree of improvement is probably too small to show meaningful conclusions.

4 Fitting Σ_0

5 Polling data

Furthermore, access to anonymized browsing history from Interet Service Providers (ISPs) may allow us to uncover latent sentiments that may be interpolated at poll-time. This will be further expounded in Model Selection.

6 Models

6.1 Browsing History Incorporation

Our prediction results rely heavily on the validity of our posterior update, which is dominated by how informative the polling data may be. However, polling data may be noisy at best and worse misleading. In fact, each pollster has such high variance that there are pollster ratings, which Part I did not explicitly take into account. The aggregate or *overall* trend in polls is what Nate Silver explains as being of greater interest.[10] In order to mitigate the noisy nature of polls and the effects of poll-to-poll variance, given more resources, we could introduce an intermediate step between poll data and votes by introducing browsing data.

On a high level, we use browsing history B for each district and assume that given this, polling results are distributed Normally. Instead of solely relying on voting results to fit the parameters of Z, we now introduce a tertiary variable through which we can (1) potentially use to directly compute $Y \mid B$ or, in our case more parochially, (2) easily derive the Σ_Z by calculating $\widehat{Z} \mid B$ and calculating the variance of the newly acquired live poll against our $\widehat{Z}$.

The method (Figure 5) entails the use of historic polling data and browsing history to fit a classifier function in each district that predicts proclivity toward a party. Given ISP's data on how a registered Democrat or Republican browses during the same time frame in which a poll is conducted, we can fit parameters of a function as simple as a regularized regression or even a neural network to classify browsers as voters (Figure 6). Model selection could depend on how we record browsing history; if we filter down to a few key websites and monitor voters' respective traffic, a simple regression model may be sufficient, while if we chose to incorporate a large number of features, we may choose a neural network of the form

$$H_m = \sigma(\alpha_{0m} + \alpha_m^T B), \text{ where } m \in [1, M]$$
$$Z_k^{(j)} = \beta_{0k}^{(j)} + \beta_k^{(j)T} H, \text{ where } k \in \{0, 1\}$$

[10] Silver argues that both macro and micro trends from polling data are important–the former being the aggregate poll result and the latter being the pollster ratings. More information at fivethirtyeight.com/features/which-pollsters-to-trust-in-2018

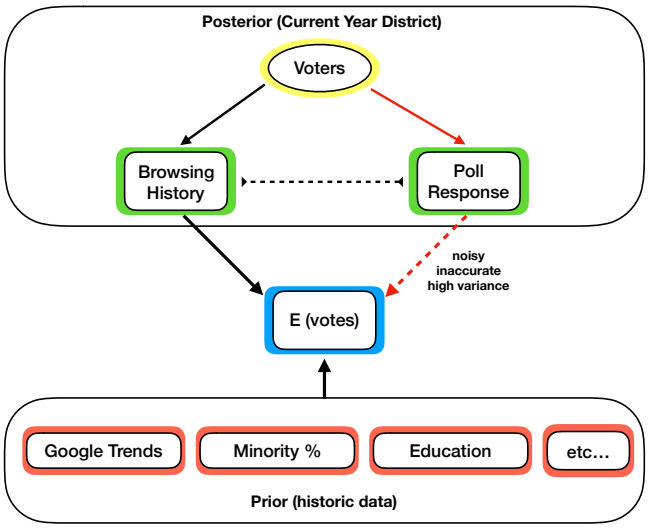

Figure 5: Diagrammatic view of incorporating browsing history as an alternative to poll-based posterior update.

where the activation function that introduces non-linearity $\sigma(\cdot)$ could be a sigmoid, a Rectified Linear Unit (ReLU), or Gaussian radial basis function; H is the hidden layer(s) with M units; and $Z_k^{(j)}$ is the aggregate polling result of district j for party k in the timeframe during which browsing history B is collected. Using stochastic gradient descent as our fitting method, we can fit the parameters to translate browsing history to expected voter behavior or compute variance of pollsters against our predicted $\widehat{Z}$. An obvious drawback of this model is the difficulty in access to historic browsing and polling data.

6.2 CANTOR

7 Discussion and conclusion

References

Agresti, Alan. 2015. *Foundations of linear and generalized linear models*. John Wiley & Sons.

	Fox	Vox
Sam	2	10
Jane	4	6
Sally	6	9
Jim	12	1

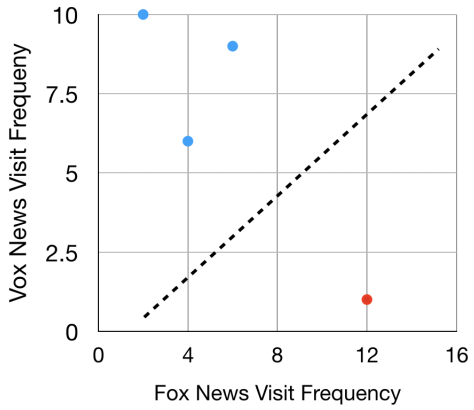

Figure 6: Example of a simple, 2-feature, unnormalized browsing history to fit a classifier for individuals to predict poll results of 75% voting Democrat.