Polarization and Voting Patterns in the United States Congress

Stephanie Herron, Cristina Lange, Seth Ringger

April 2020

Abstract

In order to understand how Congress runs, why it is becoming more partisan, and how to introduce bills that are more likely to have bipartisan support, we performed several machine learning methods on data from Congress. These include dimension reduction, clustering, and classification algorithms. Clustering analysis shed light on how members of Congress and bills are related and grouped. Results indicate that among members and bills alike, there appear to be three groupings as opposed to the expected two groups corresponding to the major political parties. We also used classification methods to attempt to predict whether bills of different types are more likely to pass. Accurately predicting whether or not Congress will vote to pass a bill or not based on topic remains a difficult problem, but results are improved when NOMINATE scores or Congress sessions are included as features in the data.

1 Motivation and Problem Statement

Because of the impact Congress has on the lives of people living in the United States as well as those in the rest of the world, it is imperative that it function smoothly. Recently, there has been a noticeable increase in polarization in our political system. By analyzing years of Congressional data, partisanship of members of Congress, and their votes on all types of bills, we hope to find information on how to help mend this gap. This includes studying how the divide in Congress is structured and what makes bills more likely to have bipartisan support.

Data on Congress is highly abundant. For over two centuries careful records have been kept for every session of Congress ever held, every Congress member who ever held office, and every bill that was ever voted on. All of this data has been rigorously analyzed over the years by many analysts, political scientists, and others. However as Congress continues to grow and evolve, many questions remain unanswered, including the two we pose here.[1]

The questions that we hope to answer through our research and analysis are these: what attributes of Congress members impact whether or not they will cooperate with the opposing party to pass a bill, and what features make a bill more likely to pass with bipartisan support? Our first goal is to use clustering algorithms and dimension reduction techniques to analyze the polarization of Congress. We use the same techniques to look at bills and find out whether there are similar divisions in the kinds of bills being proposed.

2 Data

We used data from UCLA's Voteview[2], which is maintained by the UCLA Department of Political Science and based on official records from the United States Congress. Because Congress has historically kept very thorough and precise records of Congress sessions and demographics, we believe it to be very reliable.

We used three main datasets: *Congress Roll Calls*, *Members Votes'*, and *Members' Ideologies*.

1. **Congress Roll Calls**: Information on every bill in every Congress. Features include Congress number, bill number, the vote results, and the ideological trends (NOMINATE metrics) of these votes. Also included are text labels given to the bills, found in 'clausen_codes', 'peltzman_codes', and 'issue_codes'. [2]

2. **Members' Votes**: Information on how every Congress member has voted on every bill in every Congress. Members are indexed by their ICPSR ID numbers, which is a universal identification system for Congress Members. This dataset also used the NOMINATE measures for both the bills and the members to calculate a likelihood that the member would have voted that way. [2]

3. **Members' Ideologies**: Personal/ideological information on every Congress member ever. Includes basic biographical information (state, district, party, name) and ideological scores for members of the selected Congresses. [2]

One of the most interesting metrics included in the original datasets was a measure of the political ideology of each Congress member and scores each bill. These metrics followed the NOMINATE scaling method. NOMINATE stands for Nominal Three-Step Estimation and is a well-known continuous-scale metric developed by political scientists. According to Wikipedia, "the first dimension is the familiar left-right (or liberal-conservative) spectrum on economic matters. The second dimension picks up attitudes on cross-cutting, salient issues of the day (which include or have included slavery, bimetallism, civil rights, regional, and social/lifestyle issues)." [3] It was unclear how this measure is constructed, and later in the project we will look at alternatives.

3 Ethical Implication

As noted above, there is an abundance of data concerning how Congress votes, who votes, and when Congress votes. We aim to predict when someone will vote with their party and the features of a bill that will make it more able to pass. Without proper procedures and method, our own biases can slip into our models and results. Our view can skew what we end up getting from the data. Because of that, we aim to be very careful with how we feature engineer the data and how we interpret our results. We are obligated to report what we find, regardless of whether its the result we wanted.

In analyzing Congressional bills, we make the effort to avoid bias and discrimination. While deciding what features to add or drop, our algorithms must take into account discrimination inherit in those features and try to avoid it. They should also suggest legal practices and not suggest something illegal.

4 Polarization of Congress: Methods, Results, and Analysis

In an attempt to answer our first question we engineered sub-datasets from the *Members' Votes* and *Members' Ideologies* corresponding to each Congress. These sub-datasets consisted of how the individual members of that Congress voted on each of the bills that came before that Congress. This took the form of an array where each row corresponded to a member of Congress, each column corresponded to a bill, and each entry in the array was the corresponding vote, simplified to yea (1), nay (-1) or vote withheld (0). A target was constructed from party affiliation, which joined *Members' Votes* and *Members' Ideologies* on the member of Congress's identification. We believed our question would be answered by running clustering algorithms on this newly engineered dataset. Although the NOMINATE scores offer a base for clustering members of Congress and bills, we used the voting history of the members of Congress instead. This would offer a more interpretable analysis.

PCA, K-Means, and Spectral Clustering

As a base to start from, we looked at Principle Component Analysis (PCA). After plotting scree

plots, it was clear that two principal components were sufficient to cover a majority of the variation. We noticed that in each case for the recent Congresses, there were clearly three groups as opposed to two groups corresponding to the two political parties. When individual Congress members are plotted with with color to indicate their party, two of the groups are made up of one party each, but the third is made of members of both parties. This was true for every Congress in the recent past, with the exception of our current Congress. The same method was performed with non-negative matrix factorization with similar results. The following plots show PCA performed on two different Congress sessions, and one NMF analysis.

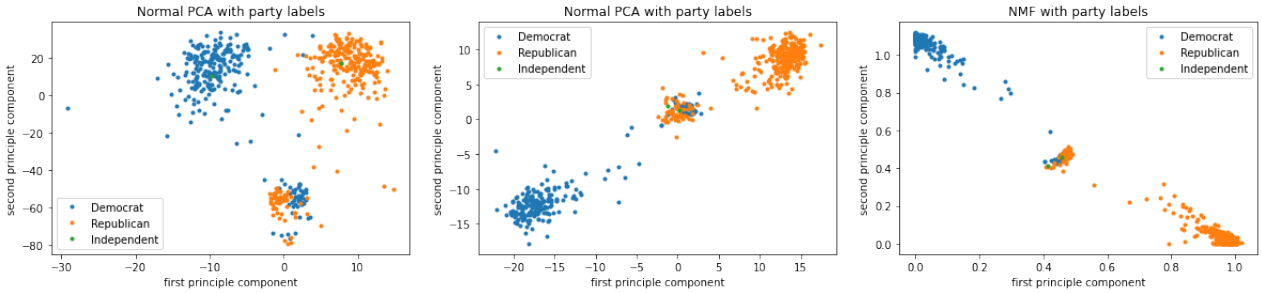

Figure 1: PCA and NMF Results on Congressmen

This suggests that there is a bi-partisan sub-group in Congress that works with each other. Although our data does not contain information about who these members of Congress are (details about their views, background, etc.), further research could show what factors are fundamental for this third, seemingly cooperative group. We used clustering methods such as k-means and spectral clustering to try and evaluate these three clusters. K-means found three clear groups each time, but spectral clustering was often able to differentiate between parties, even within the third group, suggesting that the cluster may not be as significant as we originally thought.

Non-Negative Matrix Factorization (NMF)

We decided to study the principal components themselves. A non-negative matrix factorization gives an interpretable result here since the data was engineered to be only voting for or against the bills. Features (representing bills here) with large weights in the principal components are important, but we decided to see if any topics were disproportionately represented in the principal components. To do this, we regressed over bill topic. We found that certain topics were more useful in determining whether a bill feature was important in the principal components.

Notice how the non-negative matrix factorization above groups different parties strongly in the corners. It is clear that having a high score in one of the principal components indicates strong party alignment. Thus, voting a certain way on the identified bills will group you with a given party. Thus, the topics identified with the regression are likely polarizing topics. This analysis identifies the polarizing topics in bills. According to this analysis, top five polarizing topics include Agriculture, Civil Liberties, Foreign and Defense Policy, Government Management, and Social Welfare. We further tried to see if there were certain bill topics that were indicative of being in the third, cooperative group, but with little success.

Reverse PCA

The next step was to try doing a PCA but to switch around which were the features and which were the points of data. This time, we looked at bills as the point of data and the Congress members who voted on them as the features. This led to a different view of the problem that grouped bills instead of Congressmen.

Recall that in the previous analysis, the plots indicated groups of members of Congress, and the regression indicated the importance of bill topics. Here, the plots indicate groups of bills, and our analysis of the principal components indicate which members of Congress are instrumental in the bill being placed where it is. When we analyzed the principal components, there was a clear association between party and importance in the principal component. Thus we can conclude that being in the bottom right or top left corner indicates being a partisan bill associated with that party. Generally, the NMF clustered better than PCA here. Interestingly enough, there appear to be 3 groups of bills here. There is a clear third group that is in neither of the partisan corners. It appears that Congress cooperated on these bills. After being identified, these bills could be further studied.

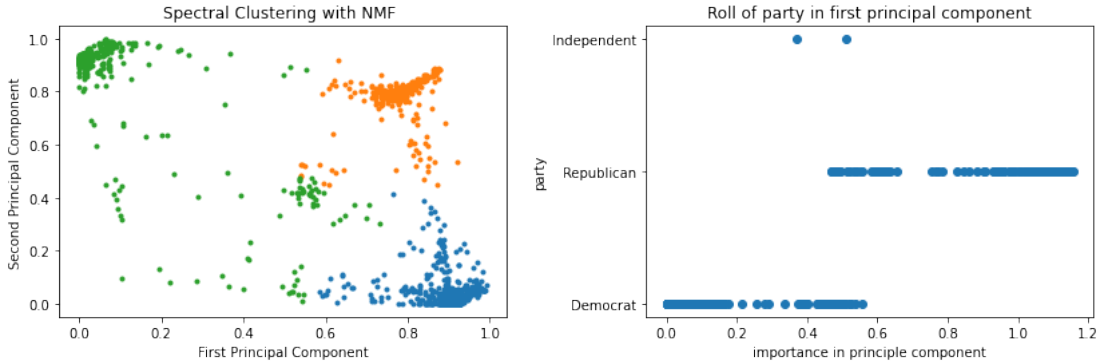

Figure 2: Clustering and Roll of Parties

Kernel PCA

Next, dimension reduction was performed on the data using kernel PCA and NMF. The first kernel to be used was the RBG kernel. Instead of putting the data into 3 groups, the kernel PCA was typically more able to split what previously, with the normal PCA decomposition, would have been the third group into two groups with clear parties, thus making four groups. This suggests that the third bi-partisan group may not be as significant as the previous analysis may suggest.

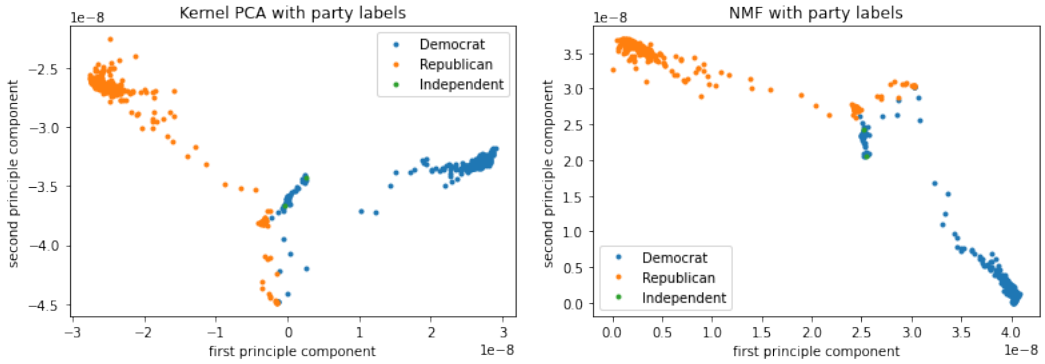

Figure 3: Kernel PCA and Clustering: Members of Congress

When we grouped bills as opposed to members of Congress, further interesting patterns occurred. There are clearly three groups, but they are harder to interpret and the lines are not as clear between the groups as with the PCA analysis. Both k-means and spectral clustering were used to group the bills, although only k-means is shown here.

Next, we used the random forest similarity measure as a kernel. This was done by training a random forest, then putting two members of Congress through the forest, the features being how

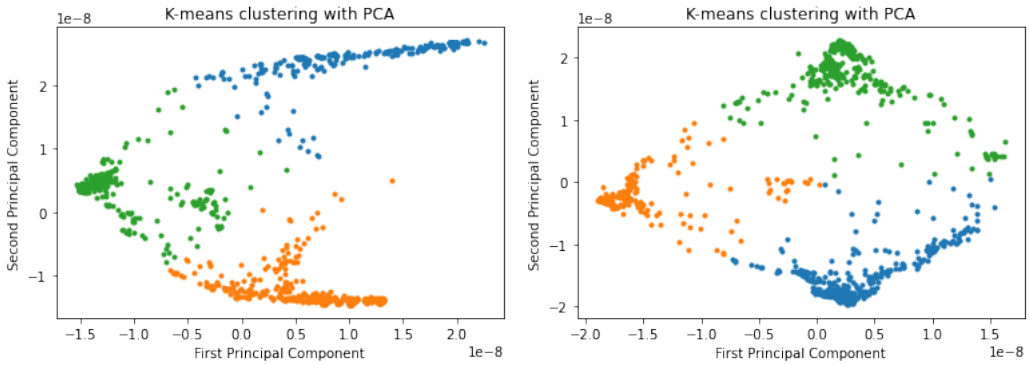

Figure 4: Kernel PCA and Clustering: Bills

they voted on individual bills. The proportion of the trees in the forest that put the two members of Congress in the same leaf determines the similarity of these two members of Congress. Next, a kernel matrix is created from this similarity measure and the above PCA and NMF analysis is then performed on the matrix. This method produced no visible third group in Congress. However, it did introduce a way to measure the polarization of a member of Congress. If they are placed at one extreme of the plot, they can be considered to be highly polarized. The few points in the middle can be identified as moderate members of Congress. Our goal was to study polarization in Congress and this gives a way of identifying and labeling strongly partisan members of Congress.

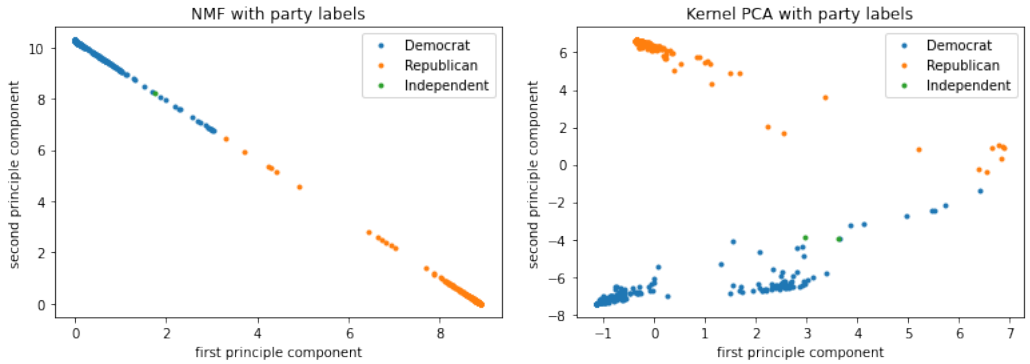

Figure 5: Random Forest Similarity Kernel with PCA and NMF

Dictionary Learning, UMAP and tSNE

As an alternative method for dimension reduction, we performed dictionary learning on the data. Dictionary Learning is an unsupervised dimension reduction algorithm, similar to Principal Component Analysis. However, the key difference is that unlike PCA, Dictionary Learning does not require that its components be orthogonal. This allows for greater flexibility in the lower-dimensional representation of the data. Dictionary Learning also encourages sparse representations of the data it is representing. We ran Dictionary Learning for both the bills and the Congressmen. Similar to Principle Component Analysis and the other methods discussed above, we saw three main clusters of Congressmen, and a distinct divide between the bills. While it was interesting to learn a new algorithm and see our results once again duplicated, we did not glean more insight from Dictionary Learning.

UMAP and tSNE gave similar results. However, they often clustered each party into two group, making four total groups, with the smaller groups from each party located close together. These

smaller groups, however, often did not form a mixed third bi-partisan group, as was found in the PCA analysis. This may suggest that the third grouping is not as intertwined as PCA would suggest, but is nonetheless significant.

5 Predicting Vote Outcomes: Methods, Results, and Analysis

To answer our second question we used the *Congress Roll Calls* dataset to evaluate outcome of a vote based on the political ideology of a bill given by 'NOMINATE', and the topics of the bill given by 'clausen_codes', 'peltzman_codes' and 'issue_codes' values. Our target values were: 'passed' representing whether the bill passed or not, and 'yea_votes' representing percentage of positive votes. Which target we used depended on whether we used classification or regression on our data.

To begin, we attempted to regress over the ratio of 'yea' votes to total votes. Reasonable outputs were ratios between 0.0 and 1.0. A score closer to 1.0 indicates higher probability of a bill passing. We adopted many of the algorithms that would later be used to classify, such as SVM, Random Forests, and XGBoosted trees. We also ran a classic linear regression on the data. We used R^2 scores to evaluate the models. However, we found our results when running classification algorithms to be both more accurate and more interpretable, so we pursued classification from here on out.

Classification

The main goal of running classification algorithms on our data is to classify a bill as being likely to pass when voted on in Congress. To prepare our data for classification, we kept features such as the partisanship of a bill (its NOMINATE dimensions) and the encoded issue codes as the training data. The classes were 1 (pass) and 0 (failure to pass), based on the yea-nay vote ratio. If the ratio of 'yea' votes to total votes was over .67, we counted it as a passing vote, otherwise we counted it as a failing vote. We recognize that whether or not a bill passes is more complicated than a simple two-thirds majority vote, but found this was a way to simplify the problem down into a binary classification problem to run and classification algorithms on.

We employed various machine learning classification algorithms, including Support Vector Machine (SVM), Random Forest classifiers, and XGBoosted trees. We also used a simple logistic regression to divide the bills between those that would pass and those that would not as a baseline to compare the other algorithms to. For each model, we performed a grid search on the hyperparameters important to the model to find the best fit. Classification results were collected using SciKit Learn's classification report method.

Below are the consolidated results for the initial classifiers.

	precision	recall	f1-score
SVM	0.76	0.75	0.71
Logistic Regression	0.72	0.72	0.71
Random Forest	0.93	0.93	0.92
XGBoost	0.93	0.93	0.93

Table 1: Average weighted accuracy for Classification Methods - all features

To NOMINATE or not to NOMINATE

When we analyzed the most important features of our classifiers, we found that they consistently relied heavily on the NOMINATE log likelihood scores of the bills. This makes sense because the NOMINATE scores define how liberal and conservative the bills are. It seems clear that we could accurately predict how a particular Congress would vote on a bill by comparing the NOMINATE score of the bill to those of the members of Congress. The top four features were always 'congress',

'nominate_log_likelihood', 'nominate_mid_1', and 'nominate_mid_2'. The most important features when looking at just the bill topics are shown in figure 6.

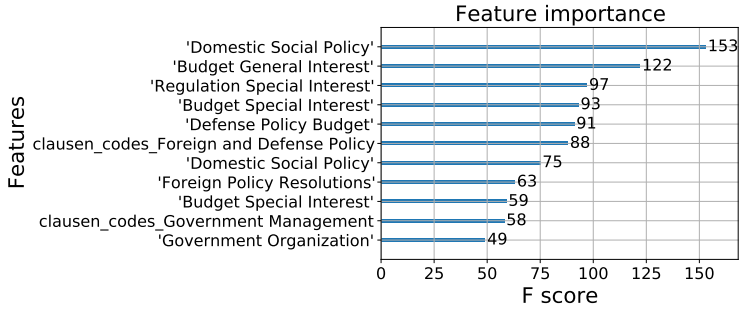

Figure 6: Feature Importance from XGBoost Model

We examined whether to keep the NOMINATE scores or not. These scores are calculated based on a formula for evaluating preferential data such as political party identification. The algorithm was developed in the early 1980s by the political scientists Keith T. Poole and Howard Rosenthal,[3] but we do not have access to the exact formulas they developed, only the scores as calculated by those who maintain Voteview.[2] Given that we did not know how these scores were calculated, we decided to use the accuracy scores calculated using the NOMINATE features as a baseline to compare our models to. From there, we wanted to know how well we could classify bills based only on their issues and topics. We ran our classifiers on the data with just the Clausen and issue codes, removing the top most important features, including all NOMINATE scores.

As we would expect, the algorithms did not perform as well when we took away several of their top most important features. They performed considerably well when only trained on their top 5 features, the various NOMINATE scores. The algorithms also ran much more quickly, because there were only 5 features to train on rather than 227 features.

	No NOMINATE			Only NOMINATE		
	precision	recall	f1-score	precision	recall	f1-score
SVM	0.66	0.67	0.63	0.73	0.72	0.70
Log Regression	0.38	0.61	0.47	0.38	0.61	0.47
Random Forest	0.67	0.67	0.63	0.96	0.96	0.96
XGBoost	0.66	0.69	0.60	0.93	0.93	0.93

Table 2: Average weighted accuracy for Classification Methods, first without the NOMINATE scores, then with only NOMINATE scores

However, we had not reached our goal of developing a classifier that could accurately predict whether or not a bill would be passed in Congress without NOMINATE scores. In an attempt to get a much better accuracy, we went back to the data. Instead of one-hot-encoding the tags, we took the strings from the names of the topics and concatenated these into a single list of tags for each bill, and put all of these lists into a single list of all the bills and their topics. We hoped that returning to the text labels would preserve some feature-bill-relationship. We then ran that list of lists through a word embedding, and then through PCA. A significant portion of the bills had more than one topic or issue assigned to them. This resulted in very sparse data of high dimension, so PCA was used to reduce the data to fewer dimensions. Finally, we passed the reduced data through our classification models. Surprisingly, the accuracy did not improve.

Returning to the original one-hot-encodings, we passed them through PCA and then through our algorithms again. This time we saw a 10% increase in accuracy. Although there was an initial jump

in accuracy with PCA, we found we could no further improve accuracy on this dataset. Our baseline was around 92%–95% accuracy and our models were performing at around 78% accuracy.

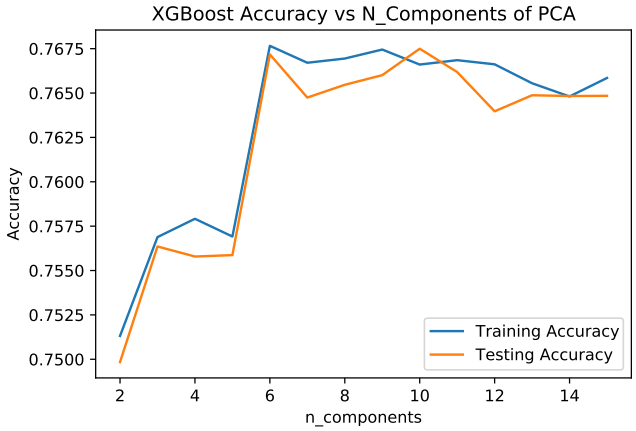

Figure 7: Change in classification accuracy as run on XGBoost for different numbers of Principal Components

Our research revealed that whether a bill is passed or not is dependent on ideology. What types of ideologies a party has will influence which bills they pass. The demographic of Congress in a particular session will affect which bills will pass in that session. While we did not have a way of measuring ideology itself, a particular session of Congress acted as a proxy for ideology, since a Congress typically votes with the majority ideology present. Adding back in the Congress session number as the feature 'congress' improved our accuracy even more than our models with one hot encoding and PCA. With this change our models performed around 80% accuracy. Surprisingly, including a PCA with 'congress' did not improve performance here.

6 Conclusion

In comparing the two sub-datasets, one sub-dataset containing the one-hot-encoded values for topic and the other containing the one-hot-encoded values with 'congress', we see a remarkable change in accuracy. As analysed above, 'congress' is a proxy for an ideology feature. The sub-dataset without 'congress' can be considered a model for identifying non-partisan issues, while the sub-dataset with 'congress' takes partisanship into account. From this we see that Congress is much better analyzed as a partisan body. This is evidence of the partisan divide.

This matches what was found when clustering algorithms were run on the political divisiveness in Congress. These results showed that there are two very strongly divided parties present in Congress as well as a third, more moderate group of Congress members. Because the Congress members in either of the two more extreme groups are unlikely to be swayed, in order for a bill to pass it must win a majority of the more moderate Congress members, those in the third group.

By running classification algorithms on the types of bills presented to Congress, we determined that certain types of bills are more likely to achieve this than others, although tracking this pattern is difficult without the NOMINATE scores of the bills. However, when trained on the NOMINATE scores, our models performed well in predicting whether or not a bill would pass. There is room for further research into the issues we have raised here.

References

[1] William Jarvis. Predicting congressional roll call votes with machine learning. 2019.

[2] Jeffrey B. Lewis, Keith Poole, Howard Rosenthal, Adam Boche, Aaron Rudkin, and Luke Sonnet. Voteview: Congressional roll-call votes database., 2020.

[3] Wikipedia. Nominate (scaling method).