U.S. Senators: A Voting Pattern Study

Ali Alami-Idrissi, Quentin Bacuet, Leandro Kieliger, Keshav Singh

I. Introduction

Predicting how Congressional legislators vote is important for understanding their past and future behavior. In this project, we explore U.S. senators voting patterns using a network-based approach. Our main goal is to use congress roll call votes results on bills, motions and resolutions to predict political stances of individuals and vote outcomes based on a restricted set of features. The latter will consist of carefully chosen votes which are designed to convey the highest information on senators voting behaviour. We will also explore whether we can accurately predict the outcome of a vote only by looking at a small, well-chosen subset of senators. We will start our study by presenting the dataset that we used throughout the project. Then, we will move to present the results of our graph-based data exploration. Finally, we will explain the methods used to achieve the goals stated above.

II. The dataset

Our project uses a subset of ProPublica's congress dataset [1]. It is comprised of several parts, the most important for our project being congress members information, voting positions and bill descriptions. We limited our usage of the data to the 115th congress as our research questions did not entail using data from previous congresses. However, as you will see in the next sections of this report, the processing pipeline developed as well as the techniques used for the project are immediately applicable to any given congress. Before diving into the details of the dataset, we deem necessary to provide a brief overview of how the legislative branch of the federal government of the United States works.

A. Overview of the U.S. senate

The senate is the upper chamber of the legislative branch of the federal government, the lower one being the house of representatives. There are 100 senators, two per state, that are elected for terms of 6 years. Senators mainly vote on bills, resolutions, motions and nominations. Bills and resolutions are pieces of legislation that are proposed and, if passed by both chamber, are sent to the president for signature. Motions on the other hand are simply propositions to bring certain subjects to consideration. Most votes happening in the senate happen ”by voice”. This means that agreement or disagreement are expressed orally and are not recorded. Subjects for which no clear majority is available will lead to a “roll call”, where the position of each individual senator will be recorded. This is the data we have at hand.

B. Features description

For the purpose of this project, we selected 96 senators from the 115^{th} congress and collected their political affiliation (i.e republican, democrat or independent) as well as their voting positions on a selected set of roll calls. The selection comprises motions, resolution and bill passages. This represents a total of 129 votes whose distribution in each category is summarized in table I. Note that deviations from the usual 100 senators are due to retirees and newcomers for which we do not have a full voting history.

Vote type	Number of votes
On the Resolution	25
On Passage of the Bill	18
On the Motion	86

TABLE I: *Number of votes kept per category*

III. Exploratory Graph Analysis

Our dataset does not have a direct graph representation. It is however possible to “artificially” build such a representation by means of similarity measures. We represent nodes as senators and link nodes by an edge whenever senators are similar in terms of their features. The features are described in II-B. Each opinion is transformed into a numerical value: 1 for yes, -1 for no and 0 whenever the value is invalid or the senator was absent during the vote. Votes are then compiled into vectors for each senators and similarities $\delta(x, y)$ between pairs of senators $\mathbf{x}$ and $\mathbf{y}$ are computed using the cosine similarity measure:

$$\delta(\mathbf{x},\mathbf{y}) = \frac{1}{2}(1+\cos(\mathbf{x},\mathbf{y})) = \frac{1}{2}(1+\frac{\mathbf{x}^\top\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|})$$

The value of $\delta(\mathbf{x},\mathbf{y})$ will therefore be close to 0 for senators that hold diametrically opposed viewpoints while a pair of senators that tend to share the same opinions will result in a high similarity value. We build two different graphs using this similarity measure for reasons that will become clear in section IV. The first graph takes into account only resolution and bill votes and will be used for the influential votes selection explained in section IV. The second graph also includes motion votes and will be used for the analysis done in this remaining section.

We sparsify both graphs by removing edges with weight below .25 and keep only the 20 edges with highest weight per node. The reason behind this design choice is that without imposing limits on the maximum number of neighbors, two clear clusters emerge immediately: republicans and democrats. The separation is so sharp, that it is not possible to sparsify further the graph in hope of uncovering community structures within parties without disconnecting them completely. Indeed, with a sparsification threshold of 0.75, the graph is already disconnected while the average clustering coefficient remains well above 0.9. The result of the second graph construction is illustrated below in figure 1.

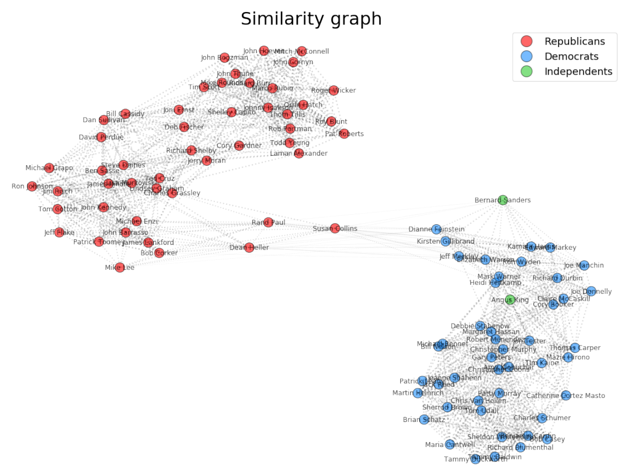

Fig. 1: *Similarity graph of U.S. senators from the 115th congress*

The degree distribution is, as expected, highly skewed towards the maximum, 20 (see figure 2, but the average clustering coefficient is now 0.64 with a graph diameter of 6.

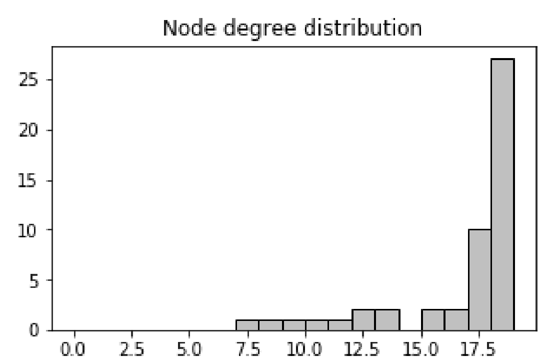

Fig. 2

IV. Finding senators that match your ideology

Our first research question was to identify the senators that represent as accurately as possible the political stance of a citizen lambda. Doing so is quite straightforward as it only requires building a new feature vector corresponding to his/her opinions on a series of questions. Ideally, those questions should be identical to what senators have already voted on. However, it would be highly impractical to ask a citizen's opinion on all of the selected votes. This is the reason why we decided to reduce that number to only 3 features. As we mentioned earlier, we will use for this part the graph which was built by only considering bills and resolutions votes. The main reason behind this choice is that motions requires in-depth knowledge about law texts. They are also often interrelated which makes their interpretation even harder.

Doing so creates an immediate challenge. Indeed, most vote topics will be similar which means that asking twice a citizen's opinion on similar subjects will not give us much additional information. To maximize the "entropy" of his answers, we chose to build a similarity graph with nodes being individual roll calls (votes on bills and resolutions) and edges representing how similar they are. This will effectively group roll calls by their "electoral basis". the sparsification process is done here by limiting the maximum neighbors number to 15 and removing edge weights below 0.5. Hence, in order to select the 3 initial votes to consider for our embedding, we designed 5 different algorithms that leverages the new votes similarity graph and its feature matrix to accomplish this task. Each algorithm is described below:

1) **Variance maximization:** The most straightforward way to find influential votes is to look for the ones that generate the highest variation in voting positions. For this purpose, we find votes feature vectors which yields the highest variance.
2) **Intra-party variance maximization:** Senators from the same political party tends to have similar voting positions. This means that method 1 might end up only capturing votes for which democrats and republicans disagree. A more robust way to identify influential votes is to look for the ones that creates the highest division inside the same party. This can be easily done by looking for the votes feature vectors that maximize the inter-party variance. Formally, let x_i be the feature vector for roll call vote i, SD the set of democrat senators and SR the set of republican senator. The mean inter-party variance for vote i is computed as:

$$\text{InterVar}_i = \frac{Var(x_{i\{j_1,j_2,j_3 \ldots j_n\}}) + Var(x_{i\{k_1,k_2,k_3 \ldots k_n\}})}{2}$$
$$\text{for } j_i \in SD \;\; \& \;\; k_i \in SR$$

We then select vote i such that:

$$i = \text{argmax}_i(\text{InterVar}_i)$$

3) **Minimize the number of neighbors in the votes graph:** An influential vote can be defined as a vote which is dissimilar to other votes in our dataset. Since our vote similarity graph is sparsified in such way that we only keep edges with high weight. We expect that an influential vote has a very small number of neighbors. Hence, the 3 initial votes can be found by simply

selecting the nodes with the lowest degree in our vote similarity graph.

4) **Finding the closest points to the bills clusters centers in the votes graph:** The graph that we constructed in the beginning of this section already permits to visually identify three clear clusters of voting subjects. Such votes can be found by embedding their corresponding nodes in the 2-dimensional space spanned by the eigenvectors of the votes graph Laplacian (i.e the ones corresponding to the smallest two non-zero eigenvalues). The clusters center can be then identified using the K-Means algorithm. Finally, to obtain the influential bills we simply take the node (bill) that is the closest to the center of each cluster.
5) **Select the most central nodes in the votes graph:** An influential vote can be defined as a vote which is related to most of the other votes. Hence, a simple approach for finding a such votes is to select the ones with maximum degree in our votes graph. However, this measure will give an equal weight to all the graph links. One could argue that a "highly connected votes" should diffuse a fraction of its importance to his neighbors. That is, we would expect that a link that relates a highly connected vote in our graph to another similar vote should have more weight than a link which relates a weakly connected vote to another one. This importance metric can be computed using a well known graph centrality measure called PageRank [7]. Hence, using the above definition, we can select the 3 most influential votes by keeping the ones that maximizes the PageRank score.

The 5 methods described above were evaluated by measuring their predictive power. That is, for each senator, we kept the 3 selected votes as measured voting positions and minimized label variation across edges by solving the optimization problem described in section VI to predict his position on the remaining votes. The final score for an algorithm is simply the mean accuracy obtained at the vote prediction task. Table II shows the scores obtained by each selection method and confirms that the clustering approach (Method 4) is the best performing one. A more detailed analysis of this benchmark can be found in our project notebook.
The selected roll call votes are illustrated in figure 3 and a brief summary of their content is shown below:

1) **Countering Iran's Destabilizing Activities Act of 2017**[4]
This bill directs the President to impose sanctions against any person that materially contributes to Iran's ballistic missile or weapons of mass destruction programs.
2) **Department of Defense and Labor, Health and Human Services, and Education Appropriations Act, 2019 and Continuing Appropriations Act, 2019**[5]
The bill provides $606.5 billion for the Department of Defense base budget for the fiscal year (FY) 2019, which is an increase of $17.1 billion above FY18 levels. This funding level is consistent with the National Defense Authorization Act as well as the recently enacted budget agreement.
3) **Disapproving the rule submitted by the Department of the Interior known as the Stream Protection Rule.**[6]
Disapprove a rule that attempted to reduce the environmental impact of coal mining by establishing a buffer zone rule blocking mining within 100 feet of streams, and imposing stricter policies that required companies to restore land to pre-mining conditions.

Method	Mean accuracy
Variance maximization	0.605
Intra-party variance maximization	0.598
Number of neighbors minimization	0.602
Clusters centroids selection	0.911
PageRank score maximization	0.877

TABLE II: *Mean accuracies obtained using transductive learning evaluation on the 3 initial votes*

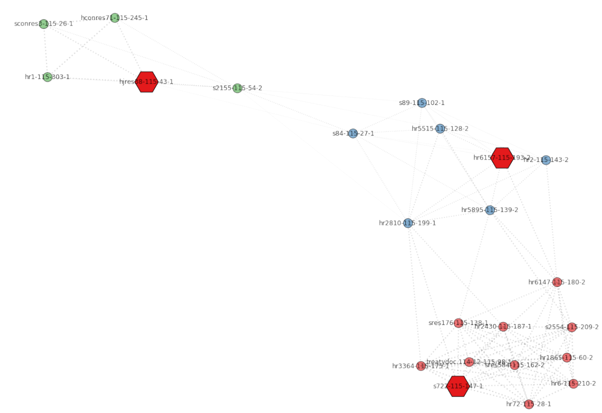

Fig. 3: *Result of bills and resolutions selection*

We see that among the selected votes, the third one is a divisive topic which can clearly define one's political orientation.
Now suppose a random citizen is asked to vote on the three subjects presented above. His answers will influence his position in the similarity graph. In order to study this effect, we embedded every node of our graph in the 2-dimensional space given by the Laplacian eigenmap [2]. Formally, let L be our graph Laplacian and U the matrix containing its eigenvectors as columns. for every senator i, we define its coordinates in our embedding as $U_{\{1,2\}i}$. Indeed, figure 4 shows four different combinations of votes and how they affect the final embedding. Note that the third subject was a divisive one. Republicans won 54-45. This explains why one's position on this subject, everything else being equal, flips the embedding of the new point from the republican to the democratic side as shown in the two higher plots. The first and second subject were less

divisive; they gathered bipartisan support. As such, individuals that do not vote according to the majority, see their embedding being moved toward the center of the spectrum as if they were “outsiders”.

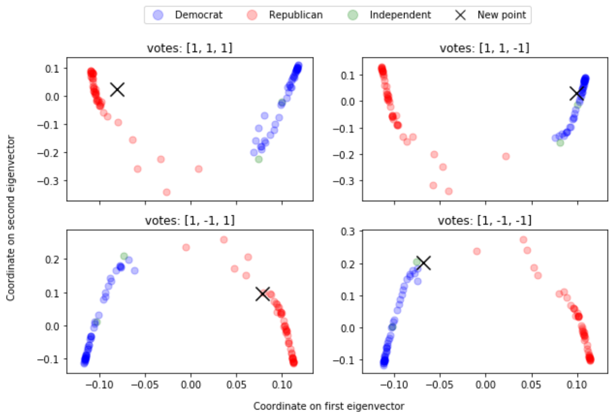

Fig. 4: *Embedding results from a restricted set of features*

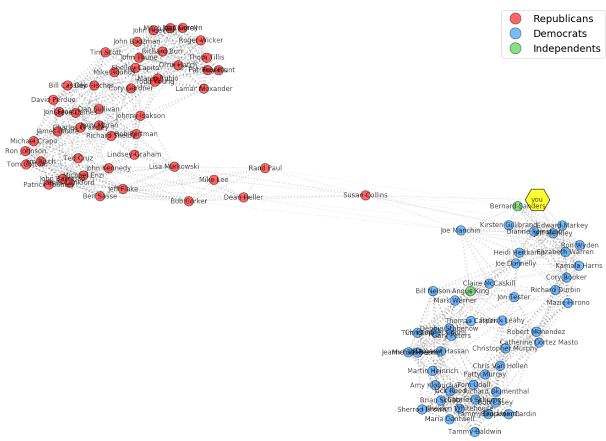

Fig. 5: *Graph when voting position is [No,Yes,No]*

V. Transductive learning: predicting votes based on similarity

A. Predicting voting positions based on similar senators

Vote positions of each senator can be seen as components of a signal z that can be overlaid on the graph. To answer our second and third research questions, we will perform transductive learning. This approach consists of fixing a certain set S of nodes from which we ”measure” the real signal y. That is, we record their voting position on a particular subject. We then aim to predict what would be the label associated with non-measured nodes in the graph, i.e, the votes of the remaining senators. To do so, we minimize the following quantity:

$$\|\nabla_G z\|_p^p = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\sqrt{W_{ij}}|z[i]-z[j]|\right)^p$$

While satisfying the measurement constraints $z[i] = y[i]$ for $i \in S$. When $p = 1$, this amounts to reducing the total weighted label variation across nodes that are connected. A simple exercise here is now to predict the voting position on unseen votes for our hypothetical citizen which we have integrated in the graph shown in fig 5.

Vote ID	Predicted vote
H.Con.Res. 71 115th 224-1	YES
S. 2155 115th 53-2	NO

They arguably make sense as the first one is a proposal by Mr. Bernard Sanders for limiting the tax cuts for the wealthier population. Since we have been embedded in a ”leaning democratic” position, we are definitely more likely to vote yes on this particular issue than no. The second one relates to a lengthy text which in essence relieves regulation on the banking sector in place since the crisis of 2008.

VI. Predicting vote outcome based on swing votes

We can however go further and ask the following question: ”Can some specific senators be used to predict the outcome of a vote ?”. The specific senators we have in mind are so-called ”swing senators” as they tend to vote less in concordance with their party. This translates into senators from one party that have connection to senators from the opposing party in the similarity graph. In partisan issues where simple majorities are required, they tend to be decisive. Recall the events of early October 2018. Brett Kavanaugh, now an associate justice of the supreme court of the United States, won a tight vote 50-48. The decisive senators at the time were Joe Manchin (D), Susan Collins (R), Lisa Murkowski (R) and Jeff Flake (R)[3]. Interestingly enough, they all lie close to the clusters junction, and in the case of Ms. Collins, at the very center of the graph.

To answer our final question, we randomly split our ”features” (votes) in two groups, a ”train” and a ”test” groups whose proportions are 80:20. That is, we build our similarity graph using the training vote positions and we test our ability to correctly predict the outcome of a vote by using the remaining testing vote positions. To assess the performance of the interpolation, we repeat the same operation with the same number of measured senators, but this time picked uniformly at random in our graph. We recorded the following values on 5-fold cross-validation:

Measured nodes	Accuracy	Std
Swing senators	0.897	$\pm$ 0.0314
Random senators	0.931	$\pm$ 0.0231

Figure 6 below illustrates the prediction results for a roll-call vote from the test set.

The 5-fold cross-validation shows that the swing senators selection doesn’t seem to predict more accurately the outcome of the votes than senators sampled uniformly a random on the graph. We propose two explanations for this phenomenon. The first one is that picking swing senators is better when trying to

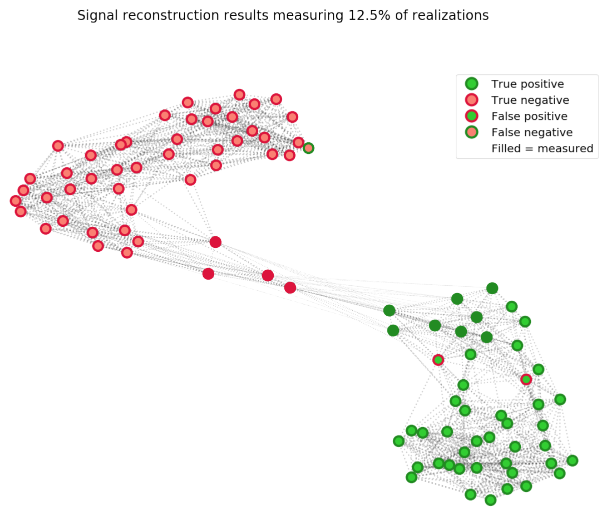

Fig. 6: *Label variation minimization from measured nodes. Example on a vote from the test set.*

predict the outcome of tight votes. These are the votes where each senator votes along party lines and the overall outcome is almost perfectly balanced. On the other hand, randomly picked senators could be really good at predicting votes where the pro/against distribution is less in balance.

We can indeed verify that there are less tight votes than non tight ones. For each vote we compute the sum of each senator's position (1 if Yes and -1 if No). Therefore, if this number is close to 0 then there was no clear majority (almost the same number of senators voted Yes and No) and the further it is from 0, the larger the majority. We can see this distribution in figure 7. Even though there is a spike around 0, there is less indecisive votes. If we restrict ourselves to votes such that their sum is between -5 and 5, we can count 57 out of 129 of them which is not a majority.

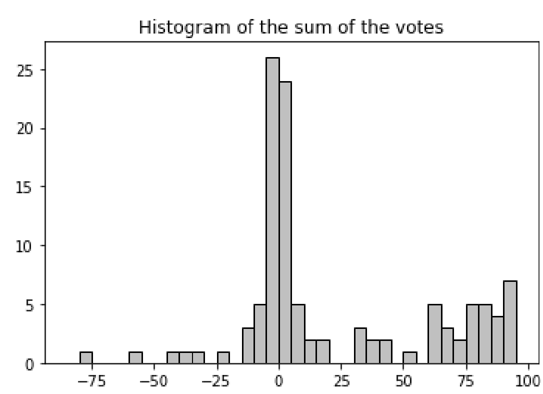

Fig. 7

Another factor that could explain our results is the fact that although swing senators might have a significant influence during tight races, this might be more the exception than the norm. Indeed, only a handful of votes result in unexpected side switches. If those situations are rare, then the mere shape of the similarity graph is already enough to provide us with a good estimate of where the vote separation will be located. Finally, by sampling uniformly at random on the graph, we are more likely to "catch" unexpected votes from a subgroup inside a party cluster than if we were limiting our measures to the clusters junction.

VII. Conclusion

In this project we have used a network-based approach to extract and identify voting patterns among U.S. senators. Semi-supervised spectral clustering allowed us to identify different categories of votes based on their "electoral basis". We defined several metrics to identify the subjects that would provide the most information about someone's political affiliation. This allowed us to reasonably integrate new individuals in the similarity graph despite the extremely small amount of features we decided to allocate for them.

We tried to predict the overall outcome of a vote by means of transductive learning. By minimizing weighted label variation along edges of the similarity graph, we attempted to interpolate the voting positions of senators from a subset of measured ones. We compared two different types of signal sampling positions. The first one, called the swing senators, were nodes connecting the republican and democratic political clusters. The second one was a simple uniform random selection. We found that using swing senators to predict the overall outcome of a vote was not more efficient than uniform random sampling, although in both cases the accuracy of the prediction turned out to be quite satisfactory.

References

[1] ProPublica.org. (2018) Congress Dataset. Retrieved from https://projects.propublica.org/api-docs/congress-api/

[2] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data repre-sentation. Neural Computation, 15(6), 1373 1396

[3] The Guardian (2018), Sam Levin, Amanda Holpuch and Paul Owen. 5 October 2018. Retrieved from https://www.theguardian.com/us-news/live/2018/oct/05/brett-kavanaugh-vote-latest-live-news-updates-confirmation-supreme-court-christine-blasey-ford-fbi-report [Accessed 12 January 2019].

[4] GovTrack.us. (2019). S. 722 115th Congress: Countering Irans Destabilizing Activities Act of 2017. Retrieved from https://www.govtrack.us/congress/bills/115/s722 [Accessed 28 December 2018]

[5] GovTrack.us. (2019). H.R. 6157 115th Congress: Department of Defense and Labor, Health and Human Services, and Education Appropriations Act, 2019 and ... Retrieved from https://www.govtrack.us/congress/bills/115/hr6157 [Accessed 12 January 2019]

[6] GovTrack.us. (2019). H.J.Res. 38 115th Congress: Disapproving the rule submitted by the Department of the Interior known as the Stream Protec-tion ... Retrieved from https://www.govtrack.us/congress/bills/115/hjres38 [Accessed 12 January 2019]

[7] Page, Lawrence; Brin, Sergey; Motwani, Rajeev and Winograd, Terry, The PageRank citation ranking: Bringing order to the Web. 1999 http://dbpubs.stanford.edu:8090/pub/showDoc.Fulltext?lang=en&doc=1999-66format=pdf