Utilizing Network Analysis to Model Congressional Voting Behavior

Janice Lan
Stanford University
Email: janlan@stanford.edu

Mengke Li
Stanford University
Email: hello.world@cs.stanford.edu

Suril Shah
Stanford University
Email: surils@cs.stanford.edu

***Abstract*—In this paper, we developed mechanisms to use network relationships between congressmen to predict their congressional voting behavior based on others' voting behavior. We used multiple networks that are summarized form different sources such as historical voting data of congressmen and social network relationships between congressmen, to generate features for our machine learning model that predicts voting behavior. At the same time, we used graph properties to identify a subgroup of congressmen who are likely to defect from their party's mutual opinion, and focused on the voting behavior prediction for this group. The result shows that we can make relatively precise predictions, which could help improve the efficiency of Congressional voting.**

I. Introduction

The recent US government shutdown brought the politics of Congressional voting under spotlight. The shutdown was a result of disagreement on key matters in a couple of bills such as the Affordable Care Act, which led to the Republicans voting it down and thus not allocating the budget for different government agencies. Although this is an extreme example of disagreement, each Congressional session sees thousands of bills out of which on an average of about 5% or roughly 758 bills are passed, and even this is on a decreasing trend. Such a vast percentage of rejected bills points to the diminishing productivity of Congress. We are interested in exploring the voting structure beneath the Congressional system from a network analysis point of view, in order to understand where this disagreement, or agreement thereof, stems from. If possible, we are also interested in figuring out whether our research results can predict how the Congress could be made more productive in terms of handling these bills.

From our observation, there are three main types of bills - the ones that are easy "Pass", the ones that are clear "Reject", and the ones that are equivocal. It is the last type, which we name "controversial bills", that we are interested in. These are the potential time consumers in Congressional sessions. Within this scope, we are interested in the aspect of the people who defected from their party's popular opinion, and resulted in a passed (or rejected) bill. We hope to predict the voting behavior of this group of people, so that the Congress can focus on a subset of this group to increase the congressional session efficiency. We present our review of some related papers in this area before going into more detail about our proposed approach towards modeling this Congressional voting network.

II. Relevant Prior Work

Snyder et al. [1] in their paper outline findings and approaches are quite relevant to what we are doing. Their methods are effective at estimating the extent to which party affiliation affects roll-call voting, independent of legislators' preferences. They estimate legislators' "true" preferences by looking at lopsided bills for which the parties do not try to influence the decisions of their members. This part of their work is useful in our goal of looking for defection potentials of Congressmen which clearly depends on the individual preferences of each Congressperson, and the amount of influence exerted on them through their party affiliation.

Another relevant paper is by Poole et al. [2], that attempts to construct a spatial model for roll call decisions in the Congress. Each legislator is represented by a point in s-dimensional Euclidean space, while each roll call is represented by two points that correspond to the policy consequence of the yea and nay outcomes. The spatial model infers that a Congressperson prefers the closer of the two outcomes, and the extent of this preference is expressed by a utility function. Choosing what they call "1.5" dimensional space allowed them to model the structure of the Congress for roll calls, and also account for the changes in party structures over time. Finally, they concluded that there was great stability of individual positions that allowed them to do short-term forecasting. Interestingly, they also showed that the distances between two parties have shrunk considerably in the last century. This work provides an interesting take on the problem we are tackling, by predicting roll call outcomes based on the position of legislators in the Euclidean space and their distance from the cutting line.

III. Algorithms and Approach

A. Data

In this research project, we obtained our dataset from the GovTrack website (https://www.govtrack.us/developers.)

GovTrack screen scrapes a variety of official government websites every day and makes the resulting normalized database of legislative information available for free to the

general public, both in bulk as well as through an API.

From this database, we also obtained matching Twitter handles for all these congressmen. We then generated a conflation between Congressmen and their Twitter accounts, and crawled their social network data, such as name, gender, screen name, Twitter id, list of follower's Twitter id and etc., to later calculate and generate social media inferred networks between congressmen.

B. Determining similarity scores

We modeled the data with a similarity graph, where each node represented a congressman, and each edge between two people was weighted based on how similarly they voted. The similarity score between two nodes was calculated as follows:

Let same_vote(u, v) = 0 and different_vote(u, v) = 0 for all nodes u, v.
For all nodes u and v and bill b, if u and v both voted yes or both voted no on b, then add one to same_vote(u, v). If one of u, v voted yes and the other voted no, then add one to different_vote(u, v). Ignore the pair of votes if anyone abstained or was present but not voting.
score(u, v) = same_vote(u, v) - different_vote(u, v) for all nodes u, v.
We excluded pairs of people who only voted on less than 10 of the same bills.

This gave us a graph of 749 nodes and 177713 edges. Across all edges, the average of the scores = 841.09. The average of (% same votes out of total shared votes) = 66.71%. It makes sense that there were more of the same votes than different votes because there were probably many bills with an overwhelming majority.

In order to discover patterns of defecting, we examined the graphs of democrats and republicans separately. This worked out nicely because the 749 nodes were split almost evenly: 373 were democrats, 375 were republicans, and only 1 was neither.

C. Analyzing unbalanced triangles

To further analyze congressional voting behavior using scores, we now have a network with signed edges, where the sign of edge (u, v) = sign(score(u, v)). Ideally, there would be two main distinct political parties, representing a bipartite graph, and the network should be balanced. However, some unbalanced triangles should be expected, especially due to defections. One point of interest was the distribution of unbalanced triangles: we wanted to find if there are only a few people responsible for most of the unbalanced triangles. To do this we simply generate the scores of all the edges, and then for each node, determine the number of percentage of adjacent triangles that are unbalanced.

D. Similarity threshold graph

Instead of having a near-complete graph, with each edge a different weight, we also experimented with modeling the congressmen with this method: given the scores between each pair, add an edge (u, v) to the graph if score$(u, v) > t$, for t = a certain threshold.

E. Twitter graph

Based the data collected from twitter, we generated two graphs, with one indicating the direct following relationships between congressmen on Twitter, and one indicating the implied relationship between congressmen based on the similarity between their complete follower base.

F. Classification

Our main goal was to predict how each person in a small group of congressmen will vote on a certain bill, given past voting behavior and other people's voting behavior. For each person p, we had thousands of data points, each of which represented a vote on a bill. For each of those data points, the output was the vote, yes/no. For the input, we included several features, both related to the specific person and to the specific bill:

1) Percent of people in p's party who voted yes
2) Percent of people of p's gender who voted yes
3) Percent of people near p's age (+/- 10 years) who voted yes
4) Neighbors who voted yes, weighted by the similarity score
5) Indegree influence inferred by the Twitter Direct Following Relationship graph
6) Outdegree influence inferred by the Twitter Direct Following Relationship graph
7) Indegree influence inferred by the Twitter Mutual Follower Percentage Relationship graph
8) Outdegree influence inferred by the Twitter Mutual Follower Percentage Relationship graph

To decide which people to test, we used the previous few analysis methods to find the top 10 most likely to defect and top 10 least likely to defect for each party (Democrats and Republicans), resulting in a total of 20 defecting people and 20 non-defecting people. These 20 are our "unknowns", and our features incorporate data from everyone else (ie. the 20 unknowns were removed from the graph temporarily while calculating the features).

For features 1-3, we simply parse through all the voting data (how all people voted on all bills) and count up the number of "yes" votes vs. total votes for relevant groups. Feature 4 uses the similarity scores from above, where the resulting feature = (sum of all score(p, v) where v voted yes) / (sum of scores of all edges adjacent to p).

To decide which people to test, we used the previous few analysis methods to find the top 10 most likely to defect and top 10 least likely to defect for each party (Democrats

and Republicans), resulting in a total of 20 defecting people and 20 non-defecting people. These 20 are our "unknowns", and our features incorporate data from everyone else (ie. the 20 unknowns were removed from the graph temporarily while calculating the features).

For features 1-3, we simply parse through all the voting data (how all people voted on all bills) and count up the number of "yes" votes vs. total votes for relevant groups. Feature 4 uses the similarity scores from above, where the resulting feature = (sum of all score(p, v) where v voted yes) / (sum of scores of all edges adjacent to p).

Features 5-6 are based on the Twitter Direct Following Relationship graph, where each edge has the same weight, and we count influence as "yes"$= 1$, "no"$= -1$, and features 7-8 are based on the Twitter Mutual Follower Percentage Relationship graph, where each edge's weight reflects the percentage of mutual followers between the two congressmen on Twitter, and we count influence as "yes"$= 1 * weight$, "no"$= -1 * weight$. For more details, check the Twitter graph model section.

For the situation when people don't have a currently active Twitter account, we would use only the first four features for the classification process.

We then used the Naive Bayes classifier to predict votes. We analyzed the performance on defecting data vs. nondefecting data separately. Specifically, we used the Gaussian naive bayes classifier available from the scikit library (http://scikit-learn.org/.) We also tried running the SVM with a linear kernel, as well as the Extra Trees algorithm for classification. However, their results were generally very close to the Naive Bayes classifier, and so in the interest of time and consistency, we only used Naive Bayes to generate our results.

IV. Results and findings

A. Determining similarity scores

Each congressman's loyalty to his/her party can be represented by the average of his/her score across all outgoing edges, ie. for every node u, find sum(score(u, v) for all neighbors v)/degree(u). Then we can sort each person by score. The score distributions are as shown in figures 1 and 2.

There is a similar shape for both parties: there is a large group of people who always vote similarly to others in the party, a large group who tend to defect (low scores), and a small group somewhere in between.

Based on this scoring system, we can find that the 5 people most likely to defect within their party are:

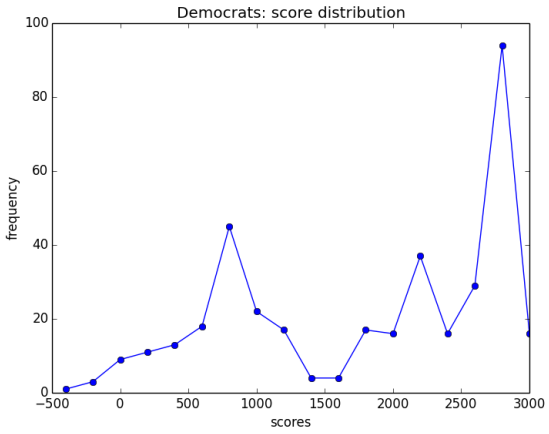

Fig. 1. Democrat Distribution

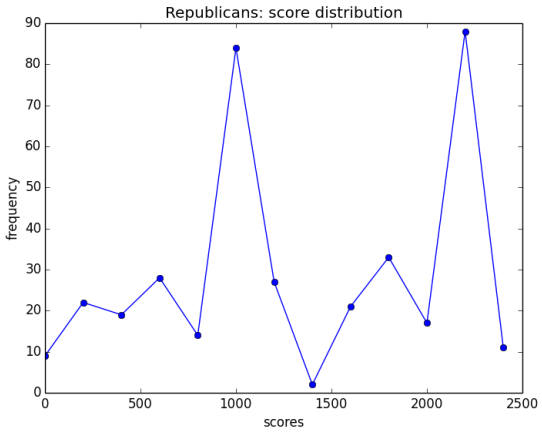

Fig. 2. Republican Distribution

Democrats: Nathan Deal, Ralph Hall, Richard Shelby, Virgil Goode, and Rodney Alexander

Republicans: Thomas Massie, Arlen Specter, Craig Thomas, Jo Ann Davis, and Luis Fortuno

B. Analyzing unbalanced triangles

We discovered that out of all the triangles between 3 congressmen, there were 47546744 balanced triangles (81.63%), 10595086 unbalanced triangles (18.19%), and 101358 that had a zero edge, which is neither balanced nor unbalanced (0.2%). This is clearly more balanced than random, but the graph is also definitely not comprised of two distinct parties. Also, the high number of balanced triangles might be partially due to the fact that there are more similar votes in general than different votes, meaning that most triangles will have all positive edges.

Within the group of all democrats, there were 8186132 (98.947%) balanced triangles, 86010 (1.0396%) unbalanced triangles, and 1110 (0.0134%) with zero-valued edges. Within

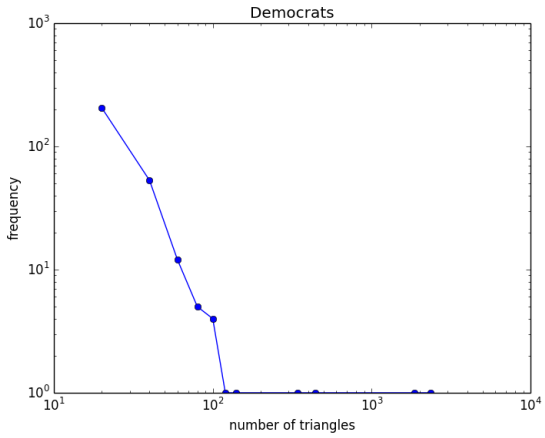

Fig. 3. Democrat Triangles

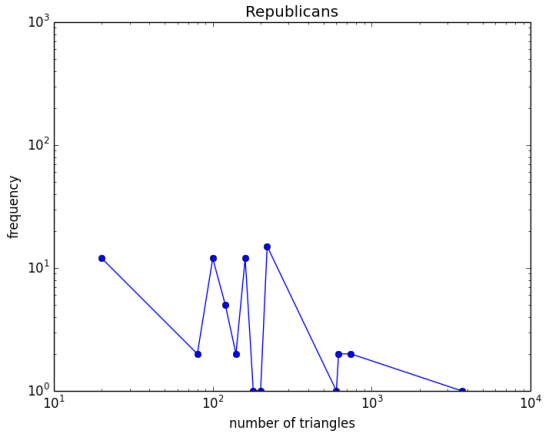

Fig. 4. Republican Triangles

the group of all republicans, there were 6840530 (99.973%) balanced triangles, 1742 (0.025%) unbalanced triangles, and 78 (0.001%) with zero-valued edges. Both of these groups by themselves were significantly more balanced than the graph of both groups together.

For the Democrats, it seems to be true that there are a few individuals responsible for most of the unbalanced triangles. Furthermore, plotting the histogram on a log-log scale gives a linear graph for most of the data, which suggests a distribution similar to the power distribution, as shown in figure 3.

The Republicans have one person responsible for a large number of unbalanced triangles. However, the power distribution property doesn't seem to hold for the Republicans. The histogram on a log-log scale is not very linear, even if we adjust the number of buckets, as shown in figure 4.

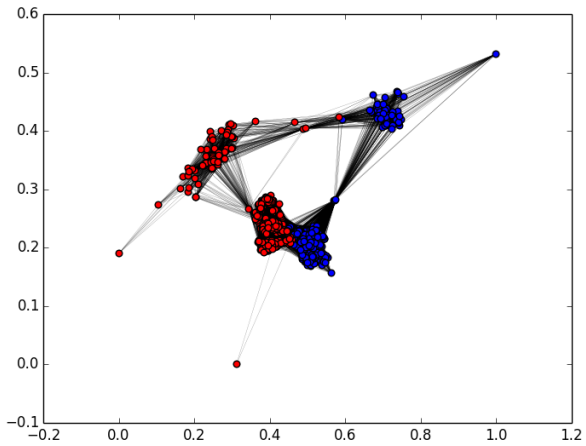

Fig. 5. Threshold = 200

By sorting the nodes by number of unbalanced triangles, we can obtain a list of people most likely to defect from a party.

Republicans: Arlen Specter, Rand Paul, Mike Lee, Susan Collins, Olympia Snowe

Democrats: Rodney Alexander, Ralph Hall, Nathan Deal, Virgil Goode, Dan Boren

C. Similarity threshold graphs

We plotted the graphs using NetworkX and Matplotlib with thresholds of 200, 1000, and 2000. The resulting graphs are shown in figures 5, 6 and 7.

In these three images, red nodes represent Republicans and blue nodes represent Democrats. If a node did not have any strong connections to anyone (no adjacent edges have weights above the threshold), then that person is omitted from the graph.

These images show that in general, there are clear clusters within the same party. However, if we zoom in to the Republican cluster for threshold = 1000 in figure 6, we can see that there are there are 4 Democrats mixed in with the red nodes. This represents Democrats who are very likely to defect from their party.

Likewise, if we zoom in to the Republican cluster for threshold = 2000 in figure 7, we can now only see 3 Democrats mixed in with the Republicans, which makes sense because the threshold for present edges is now higher, and one of them seems to be on the verge of leaving the cluster. Thus, by adjusting the threshold, this can help us

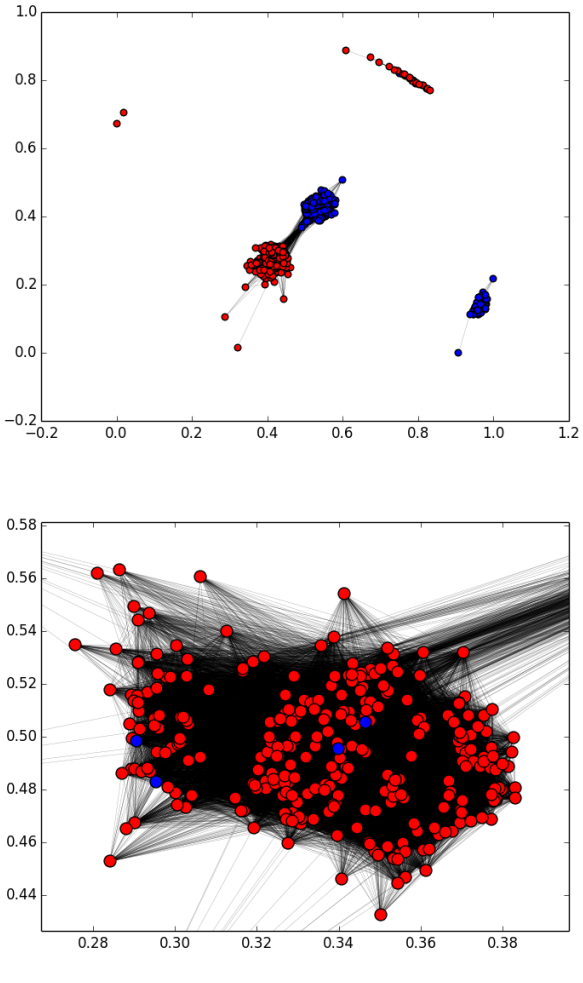

Fig. 6. Threshold = 1000, top: normal, bottom: zoomed

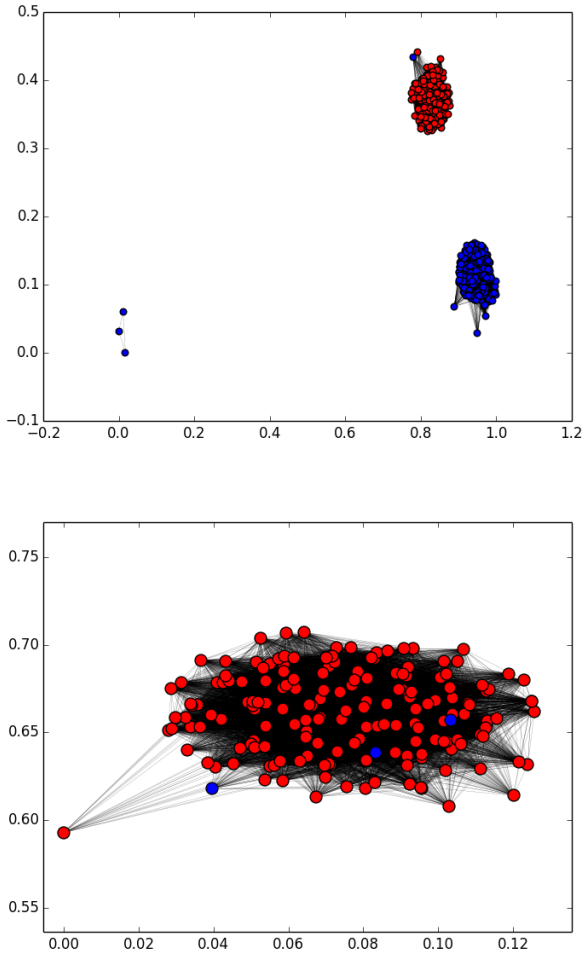

Fig. 7. Threshold = 2000, top: normal, bottom: zoomed

visually identify people who are likely to defect. In this case, some people are so likely to defect that they seem to belong to the other party in terms of voting similarity.

D. Twitter based graph

In order to expand the set of features we have for our final prediction model, as well as explore the influence from the social networks, which is considered the new generation of media, we also obtained the Twitter accounts of these congressmen, and crawled social data to generate social network relationship graphs. From the many social signals on Twitter, we constructed the the following two relationship graphs between congressmen:

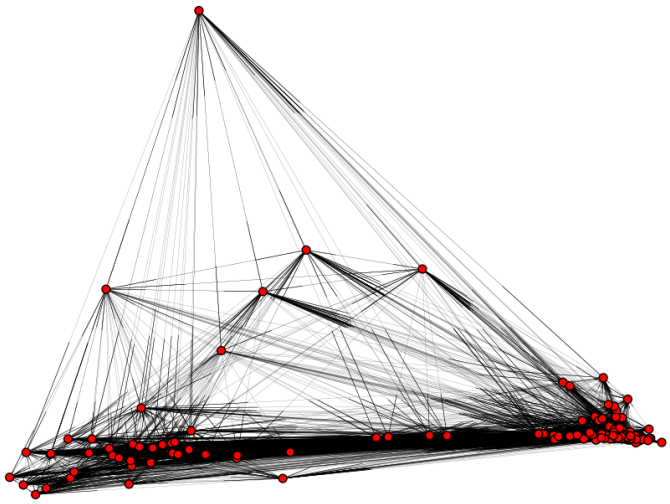

Fig. 8. Twitter follow relation graph

Figure 8:
This is a directed graph generated by the direct following relationship between congressmen. Each directed edge from congressmen 1 to congressmen 2 indicates that congressmen

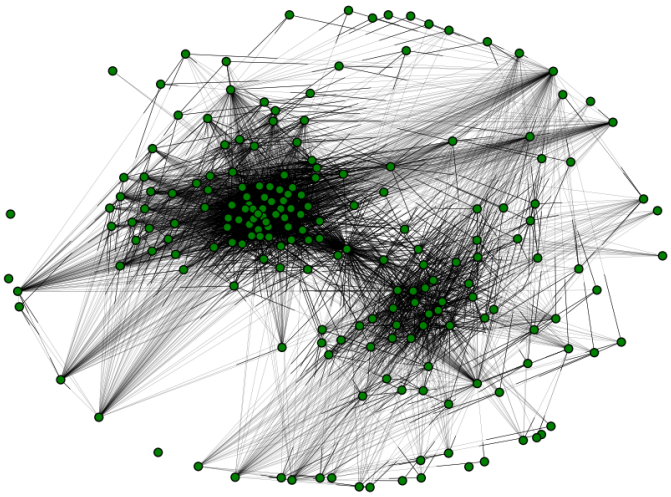

Fig. 9. Twitter mutual follower relation graph

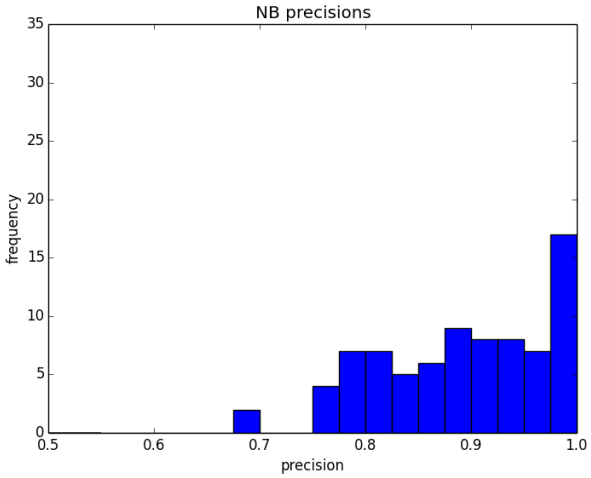

Fig. 10. Precision histograms for defects

1 is following congressmen 2 on Twitter.

The rationale behind this approach is that we believe people with closer political stance would be more likely to follow each other on Twitter, and this could help us predict the voting behavior of a given node in the network based on the voting behavior of other nodes, which represents other congressmen in the same voting session.

The graph has 328 nodes, 22127 edges, 12120 bi-directional edges. From the generated figure, we can see that although the graph is decently connected, while we can see how the majority of the nodes are separated into two different clusters. These clusters are still connected with each other, but there are much more edges in between the nodes within the clusters.

In our Machine Learning model, this graph is consumed both by calculations from indegree edges, and that from outdegree edges. We give each edge an equal weight, and for the given node, count all the indegree influence ("yes" as 1, "no" as -1) and outdegree influence and use that as a feature in our ML prediction model for the given node's voting behavior.

Figure 9:
This is a directed graph generated by an implied relationship between congressmen based on the mutual follower percentage of total followers. A directed edge from congressmen 1 to congressmen 2 would have a weight that's equal to the number of mutual followers between the two divided by the total followers congressmen 2 has.

The rationale of this approach is that we believe people following these political figures on social networks such as Twitter are more likely to follow people who also have similar political opinions on congress bills. Therefore, the mutual follower percentage between the two congressmen could act as an indicator of the likelihood these two will vote similarly on similar votes.

This is a complete graph with 328 nodes with the maximum edge weight of 0.77. In Figure 9, we have selected to show only edges with a weight that greater than 0.3. From the generated figure, we can see that comparing to Figure 8, this graph is less clustered. Although we can still see nodes clustering into two fuzzy clusters, there are more edges between clusters than in Figure 8.

In our Machine Learning model, this graph is consumed similarly as that in Figure 8. The main difference is that when we calculate indegree and outdegree influence, we count "yes" as 1 * weight and "no" as -1 * weight. We then use these as a features in our ML prediction model for the given node's voting behavior.

E. Classification

We used k-fold cross-validation to evaluate our classifier. Specifically, we used 4-fold cross-validation. Since we were generating a yes/no vote for each Congressperson, we calculated precision by comparing our prediction to the ground-truth for each bill in the test set. We obtained precision values for top Congressmen that were most likely to defect and plotted a histogram using buckets each of size 0.25%, as shown in figure 10. We did the same for the Congressmen that were least likely to defect and generated the histogram as shown in figure 11.

The mean precision value for defects was 0.894 and the standard deviation was 0.082. While for non-defects, the mean

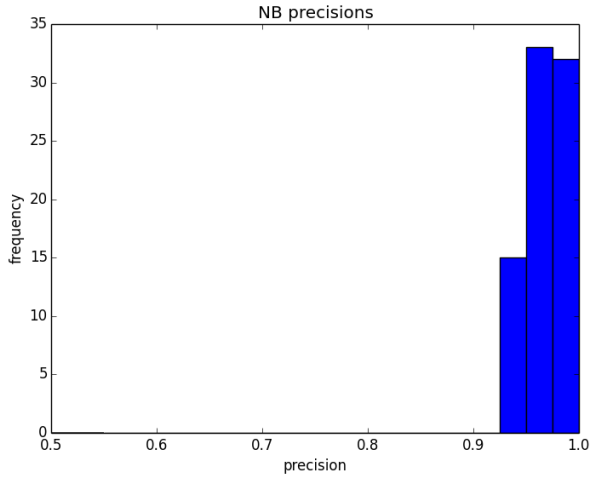

Fig. 11. Precision histograms for non-defects

precision value was 0.964 and the standard deviation was 0.014. This can also be gleaned from the histograms, as the defects histogram shows the precision values spread across between 0.7 and 1.0, while for non-defects, they are all 0.9 and above. This suggests that the defects are harder to predict correctly than the non-defects. This is understandable because the people who are most likely to defect are more volatile than the ones less likely to defect. Thus, this observation from our classifier results also backs up our methods of predicting Congresspeople more likely or less likely to defect.

V. Discussion

There are many other variations to scoring that we have yet to experiment with. For example, we can weigh bills differently, such as putting more emphasis on bills with very close votes, or on bills with statuses other than "passed". With these in mind, we can introduce more variety of features to our classifier model to improve the prediction we are making. At the same time, we are only able to explore a Naive-Bayes Classifier based prediction system, which could be a reason for the mis-predicted results by our algorithm. We could explore more approaches here such as SVM and Neural Network, to see if those improves the performance of our classifier.

References

[1] Poole, K.T; Rosenthal, H; Patterns of Congressional Voting; American Journal of Political Science, Vol. 35, No. 1 (Feb., 1991), pp. 228-278

[2] Snyder, J.M, Jr; Groseclose, T; Party Influence in Congressional Roll-Call Voting; American Journal of Political Science, Vol. 44, No. 2 (Apr., 2000), pp. 193-211