Community Detection in Political Relationship Graphs

Mickey Chao
University of Michigan
mjchao@umich.edu

Charles Wang
University of Michigan
cvwang@umich.edu

Jesse Halpern
University of Michigan
jesseoh@umich.edu

Danny Vargovick
University of Michigan
danvargo@umich.edu

ABSTRACT

Understanding politics is both difficult and critically important to the success for any country. We apply graph mining techniques to analyze the structure of the United States Congress. The object of interest is a tripartite graph connecting companies to legislators based on corporate donations and legislators to bills based on their voting records. We apply graph mining algorithms such as SimRank and Node2vec to discover underlying structure among companies, legislators and bills. Specifically, we provide quantitative evidence showing that politicians are systematically paid by companies based on the views they hold and supporting the possibility that corporate donations influence the bills that get passed.

1. INTRODUCTION

The world of politics is a relationship-driven industry that is extremely difficult for an outsider to understand fully. We posit that we can leverage our skills in mining large-scale graph data to better understand this complicated web of industry protocols and unwritten rules. There are three main goals we wish to achieve, and we approach subproblems within each goal and explore different methods to answering each question.

Our first task is to characterize the political stances of legislators based on the bills for which they vote. We first analyze a graph where there exists an edge between two politicians if they voted the same on a bill, and the weight is the number of bills that they have voted the same on. Afterwards, we analyze the bipartite graph connecting politicians to the bills for which they voted. This is useful because we want to look past politicians' often two-faced rhetoric and evaluate politicians based on the things that matter – their votes. Everyday voters do not have the time to constantly track how their representatives are voting; we can do that for them and help them be better informed voters. If a politician gets elected based off a campaign promising to be moderate, but, their votes are very similar to other politicians who are more extreme, the public needs to know this.

Our second task is to characterize the political stances of legislators based on the companies from which they receive donations. We analyze the bipartite graph connecting companies to the politicians to whom they donate and how it changes over time. Since the Citizens United ruling in 2010, companies have had the green light to donate outrageous sums to politicians. We want to know how this money is affecting our politicians and our laws, what companies make similar donations, and what politicians accept similar donations. We compare the performance of identical models on voting data and company contribution data to see which factor, votes or contributions, as more indicative of a legislator's choices.

Our third task is to characterize the relationship between companies and bills. We analyze the entire tripartite graph together to determine if there are bills that certain companies may be "targeting" with their donations. This goes back to address the question of how money affects US politics and if corporations, including nonprofit ones, should receive greater scrutiny for making political contributions.

2. DATA

We parsed and gathered two data sets: one linking organizations to politicians by organizations' donations to those politicians, and the other linking politicians to bills by politicians' votes on those bills.

2.1 Campaign Finance Contribution Data

Contribution data was found from opensecrets.org [2] and scraped into a csv format. This data pertained to the bi-annual makeup of congress from 1998-2016 and contain the top 100 donations for each Congress member during a two-year cycle of congress. Data from opensecrets.org was scraped using the python libraries urllib2 and BeautifulSoup. Urllib2 was used to fetch the webpage content from opensecrets.org. BeautifulSoup is a python library that takes an html file and has built-in functions to allow users to easily grab the wanted information. In order to gather this data, we first needed to scrape the URLs for every Congress member for every year since the IDs in the URL were in no iterable pattern. Using these scraped URLs, we then scraped information regarding each Congress member's top 100 donations.

2.2 Voting Data

GovTrack.us is a website dedicated to publishing records regarding current and previous federal legislation, as well

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

as information on both historic and current senators and congressmen. We will use their records to obtain the voting records of individual congressmen on bills introduced during that session, as well as accompanying biographical data. Data is divided into Congresses, which each contain two years of voting data. Congressmen can appear in multiple Congresses, however we will not compare congresses that span different ranges of years due to different membership between them.

GovTrack.us seperates the data regarding bills into directories that are organized by congressional sessions, years within these sessions, and bills within these years. We first collect this data in bulk, using rsync, a Unix/Mac tool for efficiently fetching files. Each directory contains a json file which contains information about the bill itself, as well as how each congressman voted on the bill. Iterating over each bill in a congressional session directory, we build a dictionary mapping each congressmen to two lists, one containing the IDs of each bill they voted 'yes' on, and one containing IDs of each bill they voted 'no' on.

2.3 Comparing the Data Source

The sizes of these two sets of graphs differ greatly. With only about 500 bills per session, the graph composed of voting data has on average about 1000 nodes and 100000 edges. We did not exclude edges from this graph in order to retain as much information as possible. The graph composed of contribution data, on the other hand, has the same number of congressmen, but about 15000 companies per graph, on average. Yet, even with many more nodes, the contribution data only has about 90000 edges above the contribution threshold of $12,000. This threshold was decided upon by finding the 90th percentile of all contributions in that session of congress, and was imposed in order to reduce the complexity of the problem and allow our algorithms to run in an acceptable amount of time. This means that the graph composed of voting data is much denser (nearly fully connected) than the graph composed of contribution data. We believe this has implications on the results received from our algorithms which are discussed later in this paper.

3. PROPOSED METHOD

In our project, we applied two algorithms, Simrank and Node2vec, to construct representations of the entities in our political relationship graphs. We initially applied Simrank and then switched to Node2vec.

3.1 Simrank

The intuition behind our usage of Simrank is that congressmen who are similar may share similar voting records and that congressmen who are similar may have similar donors. We have chosen to represent the data as a tripartite graph, with nodes representing members of congress, bills, and companies. There are no edges within the groups of congressmen, bills, or companies, as well as between groups of bills and companies.

In order to cluster the nodes, we propose first establishing a similarity measure between nodes, and then clustering these using these similarities in conjunction with a clustering algorithm. By running these algorithms on each bipartite partition of the tripartite graph, and then comparing the overlap of the cluster found within each subgraph, we will be able to observe the correlation between political donations and congressional voting records. The bipartite partitions are then collapsed into adjacency matrices. Our first adjacency matrix corresponds to the voting records of congressmen and has edges between congressmen who have voted on bills together, and these edges are weighted by the inverse of the number bills they voted identically on. We take the inverse of the number of identical votes in order to create a graph with a shorter distance between two nodes corresponding to a higher degree of similarity. Similarly, we create an adjacency matrix for the congressional contributions data where edges correspond to the inverse of the number of companies who have donated to both congressmen in the adjacency matrix. For example, if 4 companies donated to both congressmen i and congressmen j, the ith row and jth column in the adjacency matrix will equal $\frac{1}{4}$.

In Simrank, if two nodes are similar, then they will have a shorter average path length between them than nodes that are dissimilar. In our voting subgraph, congressmen who share high numbers of votes will be more similar to each other and each other's similar colleagues. On the other hand, congressmen who share fewer numbers of votes will be dissimilar to each other and each other's similar colleagues. In our donations subgraph, this means that two congressmen are similar if they have similar donors.

The Simrank algorithm works by first precomputing the neighbors of all nodes within a radius r. Let N_a denote the neighbor set of a node a, the set of all nodes within distance r of a. The algorithm then considers pairs of nodes (a, b) where $b \in N_a$[4]. It performs the iterative update

$$s_{i+1}(a,b) = \frac{C}{|N_a||N_b|} \sum_{v_i \in N_a} \sum_{v_j \in N_b} s_i(v_i, v_j)$$

We extended the traditional Simrank algorithm to account for weighted edges by modifying the iterative update to be

$$s_{i+1}(a,b) = \frac{C}{|N_a||N_b|} \sum_{v_i \in N_a} \sum_{v_j \in N_b} s_i(v_i, v_j) \cdot \frac{W - w_{a,i} + 1}{W} \cdot \frac{W - w_{b,j} + 1}{W}$$

where W is the total edge weight in the graph and $w_{a,i}$ is the weight of the edge from a to its neighbor v_i and $w_{b,j}$ is the weight of the edge from b to its neighbor v_j. In this alternative formulation, if two nodes a and v_i are connected by a large edge weight (i.e. are farther apart), then the value $\frac{W - w_{a,i}}{W}$ will be smaller and penalize the similarity. Also note that if all the edge weights are 1, this is equivalent to unweighted Simrank[5].

The upper bound on the runtime complexity is $O(|V|^4)$ if the graph is completely connected. However, this tends to run much faster because choosing a lower value of r allows us to prune more nodes out of the neighbor sets. The r value is set differently for different graphs because we have weighted edges. C, the decay constant, is set to 0.6. We also try to run the algorithm for 5 iterations, but cut it down if the program takes a long time to run on the data.

3.2 Simrank Clustering

Once we have Simrank scores, we apply a variation of K-Means clustering in an attempt to separate legislators based on their party affiliation: Democrat or Republican. Traditional K-Means uses Euclidean distance as a measure of similarity. However, our Simrank algorithm has already computed the pairwise similarities between nodes. Therefore,

we apply a modified K-Means algorithm to use the Simrank similarities to cluster nodes.

The K-Means algorithm iteratively assigns nodes to one of n clusters, where n is a hyperparameter provided upfront. Each cluster has a value corresponding to the mean of all clusters within it. To initialize the algorithm, nodes are randomly assigned to one of n clusters. The mean value is then recomputed for the cluster, and then the nodes are reassigned to the cluster with the closest mean value to itself. We continue this process until the nodes converge to a stable cluster assignment, or until the algorithm has reached k iterations. In this paper, we use a k value of 100 iterations.

The results from Simrank were indicative of there being structure in our graphs. Unfortunately, they were not clean and effective enough. Simrank gave us a 1-dimensional score of similarity, which most likely was not enough to discover clean characteristics of the data. Furthermore, we lacked domain expertise in political science and were unable to engineer additional features effectively. Consequently, we resorted to Node2vec, an algorithm that automatically learned higher-dimensional representations with which to work.

3.3 Node2vec

A second representation with which we experiment is Node2vec embeddings. Node2vec is an algorithm which applies a neural network to embed nodes in a low-dimensional feature space. The neural network learns a feature space that generally provides excellent structural information about the nodes. At a high level, Node2vec's works by optimizing the following objective function, which maximizes the log-probability of observing a network neighborhood $N_s(u)$ for a node u conditioned on its feature representation, given by f:

$$\max_f \sum_{u \in V} \log Pr(N_s(u)|f(u))$$

$f : V \to \mathbb{R}^d$ is the mapping function from nodes to feature representation and is what Node2vec optimized for. This equation is broken down and optimized using stochastic gradient ascent over model parameters defining the features f. Node2vec has been designed with a flexible biased random walk procedure that can supports neighborhood exploration in a BFS and DFS manner[6].

In our project, we find a 128-dimensional space in which to embed the nodes. We then applied algorithms such as K-Means clustering and PCA to interpret the embeddings returned by Node2vec.

3.3.1 K-Means Clustering and Shannon Entropy

First, we analyze the bipartite graph of corporate donations to legislators. We run Node2vec on the graph to learn vector representation of nodes. We apply K-Means clustering to the nodes that represent legislators and attempt to find two clusters to represent Democrats and Republicans. There few (less than 5) Independents in each of our datasets so Independents are removed from consideration. We then look up the ground truth of the legislators in both clusters and score the performance of the algorithm based on the Shannon entropy of the two sets. Specifically, the Shannon entropy of a cluster with datapoints X labeled as either Democrat or Republican is

$$H(X) = -(P(\text{republican}) \log_2(P(\text{republican})) + P(\text{democrat}) \log_2(P(\text{democrat}))$$

where P(republican) is the probability that a randomly selected datapoint in cluster X is labeled as Republican and P(democrat) is the probability that a randomly selected datapoint in cluster X is labeled as Democrat. Shannon entropy has a maximum value of 1, which indicates that the clusters are essentially random, and a minimum value of 0, which indicates that the clusters perfectly separate Democrats from Republicans.

Afterward selecting the performance measure to be Shannon entropy, we record the performance of Node2vec followed by K-Means clustering on the corporate donations graph over time and observe changes for Congressional sessions 105 through 114 (inclusive).

3.3.2 Principal Component Analysis

To interpret the neural network's embeddings, we also apply principal component analysis to the node vectors returned by Node2vec. We generally select two (occasionally one or three) principal components and project all node vectors onto the space spanned by those components and produce a scatterplot of the nodes labeled by their ground truths. These images help show what attributes each individual component, or combination of components, represents.

3.3.3 Two More Graphs

We apply the same process described previously on the voting records bipartite graph that connects legislators to the bills for which they voted and the complete tripartite graph that connects companies to the legislators to which they donate and legislators to the bills for which they vote. Then, analyze differences in performance on different datasets.

3.3.4 k-Nearest Neighbors

Finally, we apply the k-Nearest Neighbors algorithm to find the bills that are the closest to specific companies. There is no ground truth for whether companies target specific bills, but we report bills that the neural network found to be close to some well-known organizations and manually check if the results make any sense.

3.4 KMeans Clustering on Shared Voting Adjacency Matrix

Once we have the bipartite graph of politicians' "yes" votes for bills, we construct a new shared bill matrix representing the number of common "yes" votes between two politicians. For example, if politicians 1 and 2 have both voted yes for bills A, B, and C, the entry [1, 2] will be 3. Thus, entries of this matrix with higher values represent a higher similarity in voting behaviour between two politicians. Each row of this matrix can then be treated as a sort of feature vector for a politician where two vectors with a small euclidian distance should indicate similar politicians. We will then run KMeans on this matrix to form two clusters of politicians. This method of clustering is similar to a recommendation algorithm which suggest the closest products to a given product.

4. RESULTS

4.1 Simrank

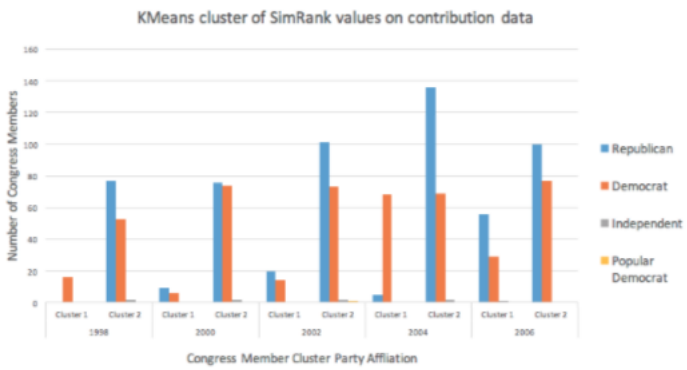

Figure 1: Results of clustering legislators by similarity on donation data.

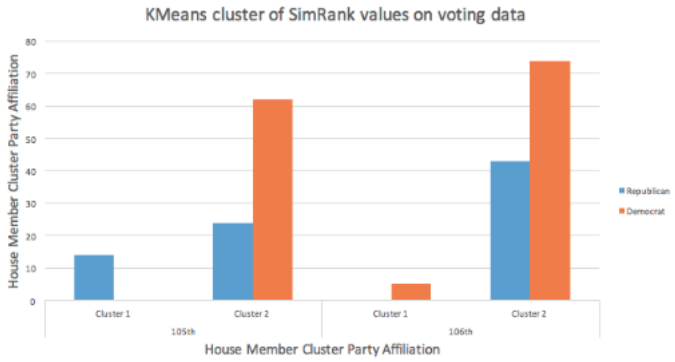

Figure 2: Results of clustering legislators by similarity on voting data.

Our initial results for SimRank and K-Means were promising. Due to time constraints, the hyperparameters for K-Means and SimRank were tuned to run efficiently rather than to completion. With these hyperparameters, we ran SimRank on both the congressional voting data set and the congressional contribution data set.

For our voting data, we found that the averages of SimRank scores for congressmen within the Republican and Democratic parties was consistently non-zero, while the average values for SimRank scores between members of the Republican and Democratic parties was consistently 0. With a small r-value for SimRank, this is expected; Congressmen who don't share many votes will quickly be pruned from the set of neighbors to explore for each node.

In addition, when running K-Means on voting data, we consistently found that at least one cluster would contain only Democrats or only Republicans. This is also to be expected; Congressmen typically vote along party lines, so with SimRank scores obtained with small r-values, we would expect there to be a very strong relationship within the Republican and Democratic parties.

For our contribution data, we ran SimRank on the entire bipartite contribution graph unlike the voting data where SimRank was run on a graph of only politicians (that was created using the bipartite voting graph). SimRank was run on congress sessions in the years 1998, 2000, 2002, 2004, and 2006. For each SimRank results we filtered down to scores relating politicians to other politicians. We then performed K-Means clustering on each of these compiled SimRank results. The clustering we found here was not as distinct as the voting data as both clusters for each year usually comprised of some amount of Democrats, Republicans, and Liberals. Session 2004 gave the best results where we saw a significant majority of Democrats in cluster 1 and a slight majority of Republicans in cluster 2.

However, K-Means tends to favor a single very large cluster in addition to smaller other clusters. It was typical for a single cluster to contain over 85% of all nodes in a given graph. This was true for both the voting and contribution data. As mentioned earlier, Simrank only returns a similarity score which provides 1 dimension of structure. We suspect this led to high bias and resulted in poor clusters.

4.2 Node2vec

4.2.1 *Initial Exploration*

We initially ran Node2vec on corporate donation data for Congressional session 113. The K-Means clustering algorithm achieved an entropy score of approximately 0.25 among Republicans and 0.16 among Democrats. Specifically, it correctly clustered 274 Republicans together and incorrectly added 12 Democrats to the Republican cluster. It correctly clustered 247 Democrats together and incorrectly added 6 Republicans to the Democrat cluster. We displayed some of the incorrectly clustered legislators, and obtained some of the following people:

- Sanford Bishop (Democrat classified as Republican)
- Jim Costa (Democrat classified as Republican)
- Tim Johnson (Democrat classified as Republican)
- Michael Grimm (Republican classified as Democrat)
- Pete King (Republican classified as Democrat)
- Chris Smith (Republican classified as Democrat)

Upon some additional research, we discovered the majority of the classifier's mistakes were for people who contradicted their party ideals. For example, according to Wikipedia, "[Sanford] Bishop is one of the more conservative black Democrats in the House" and our algorithm classified him as Republican while the ground truth stated he was Democrat.

Overall, we were surprised to find that corporate donations provided enough information to separate Democrats from Republicans with very low intra-cluster entropy. This leads us to believe that corporate donations are extremely systematic and provide information that voting records should provide. Therefore, we believe this serves as strong quantitative evidence showing that politicians are paid by corporations for voting in their specific ways.

4.2.2 *Temporal Entropy*

We also explored the classifier's performance on corporate donation graphs for different sessions of Congress. We produced the following graph:

For Congressional sessions 105-111, the performance of the classifier averaged around 0.35. Starting with session 112, the entropy dropped below 0.30 and is currently holding steady around 0.24. There was a significant increase in

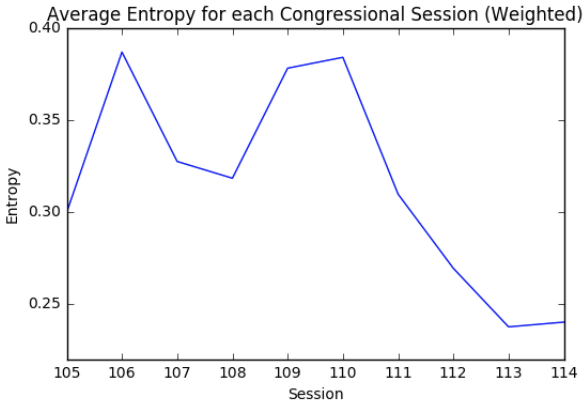

Figure 3: The entropy of clusters produced by k-Means on the node vectors.

performance (decrease in entropy) starting with the 111th Congress. During session 111, the Supreme Court passed *Citizens United vs. FEC* which allowed companies to donate an unlimited amount of money to politicians. Congress 112 was the first entire session throughout which *Citizens United vs. FEC* was instated.

Our temporal data suggests that an anomalous event affected the correlation between corporate donations and political stances of legislators. Specifically, it points to *Citizens United vs. FEC* as an explanation for the anomaly. The primary variable in the dataset before and after session 111 is the amount of money corporations are allowed to donate to politicians. This would suggest that the increase in corporate donations solidified the political stances of legislators. We may be led to believe that the increase in corporate donations "fixed" the political stances of legislators and warrants greater scrutiny.

4.2.3 *Interpretability*

Since we applied a neural network to our data, the results may be more difficult to interpret. We applied principal component analysis to show that the embeddings capture important features of the data. Specifically, we projected all the node vectors onto the first and second principal components and obtained the following plot:

As the graph shows, the first principal component is the party affiliation of each node. Negative values along the x axis represent the "Republican" characteristic while positive values along the x axis represent the "Democrat" characteristic.

The first principal component can be very useful in practice to determine the relative political moderation or extremism of different politicians. Clearly, politicians' stances on issues are a lot more nuanced than simply the binary Democrat or Republican, but it's difficult for people at home to determine which of two Democrats are more liberal or more moderate. Our first principal component can help resolve this issue. Additionally, websites like OnTheIssues that use writers to report on the relative political positions of different politicians could use this data-driven approach to aid their efforts of increasing political transparency for the general public. Upon inspection, our results seem to hold up to general public opinion. Ted Cruz is rated as more

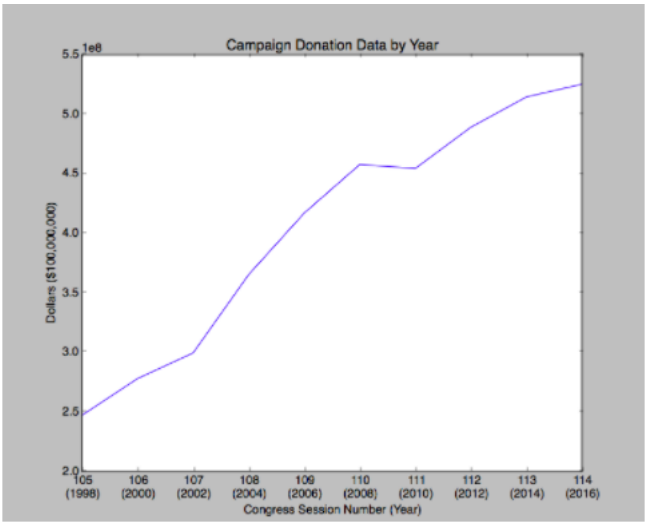

Figure 4: Campaign donations since 1998. The 105th Congress is the first session that OpenSecrets goes back to. Donations have steadily increased over time, and are now over two times what they were in 1998. However, we don't see any sort of exponential growth after January 2010, when Citizens United passes. Interestingly, we see a slight dip before Citizens United passes, potentially due to some sort of confusion or apprehension before the ruling.

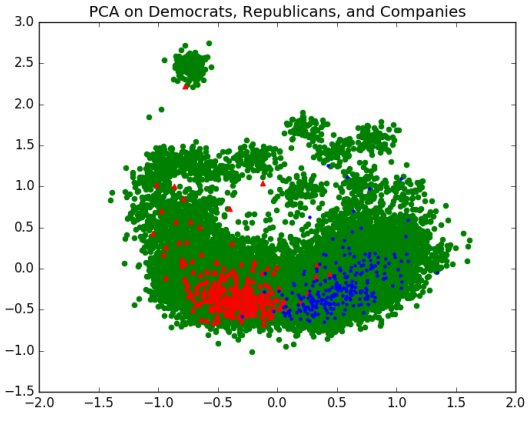

Figure 5: Results of applying PCA on the node vectors.

conservative than Marco Rubio and John McCain; McCain was one of the first and most vocal Republicans to disavow at-the-time-Republican-nominee Donald Trump after sexual assault accusations surfaced against Trump.

Alternatively, we can identify politicians who are nominally Democrats but in practice act quite conservatively or who are nominally Republicans but in practice act quite liberally. It is important that voters know that a politician acts in a way that is consistent with how they expect them to act; often voters don't have time to do any research on a candidate past looking past a party affiliation.

According to our first principal component, Chris Smith, a member of the House of Representatives representing New Jersey's fourth district, is the most liberal Republican. Smith has worked to increase New Jersey's Medicare reimbursement rates, a decidedly liberal objective. Smith is conservative socially; companies likely care more about politicians' economic policy stances than social views. Ileana Ros-Lehtinen, a member of the House of Representatives representing Florida's 27th district, is the second most liberal Republican per our first principal component, and she was first Republican to support same-sex marriage.

Our first principal component ranks Henry Cuellar as the most conservative Democrat. Cuellar describes himself as a "moderate-centrist." The National Journal has described him as "the most centrist member of the Texas congressional delegation." The second most conservative Democrat is Jim Cooper, a member of the House of Representatives representing Tennessee's fifth district. Throughout his political career, Cooper has supported the National Rifle Association, and once, the NRA donated close to $10,000 to his campaign, which was approaching the then-legal limit. Cooper has supported health care plans that do not ensure universal coverage for all Americans, and Hillary Clinton has clashed with him on these proposals at times.

After researching the backgrounds of several legislators with extreme values along the second principal component we believe the second principal component represents relative importance in the US Congress.

Politicians with the most negative values on the second principal component were all Senators, often from important states. Generally, Senators are more important than Representatives since there are significantly less members of the Senate than the House of Representatives, but the Senate and House themselves are equally important. Additionally, some of the members of the House of Representatives that were deemed most important in the second principal component were representatives from districts that contained large cities. For example, Ann Wagner was the 24th most important Congress member, and she represents St. Louis.

On the other end, the Congress member with the most positive value for the second principal component was Eni Fa'aua'a Hunkin Faleomavaega, Jr., the Delegate to the United States House of Representatives from American Samoa's at-large congressional district. Madeleine Mary Zeien Bordallo is the Delegate from the United States territory of Guam to the United States House of Representatives. Delegates are individuals who are non-voting members, so they are clearly the least important members of Congress.

We are still attempting to interpret additional principal components, although this may be difficult without domain expertise and hard-to-collect ground truths. We are very confident, though, that the principal components will represent important features of the nodes.

4.2.4 *Exploration on Voting Data and Tripartite Graph*

The voting data did not provide much signal with which Node2vec could work. When we applied Node2vec and then attempted to cluster the nodes for session 113 into two groups, the average entropy was approximately 0.99, or essentially random. We do not have conclusive reasoning as to why the voting data performed significantly worse than corporate contribution data, although we have a few speculations.

The most likely explanation is that the voting data simply contains much more noise than the corporate contribution data. We would expect corporations to only donate to relevant politicians, and thus, most of the corporate contribution data is probably relevant to our task at hand. On the other hand, for votes on bills, there may be more noise. For example, we noticed there were situations where everyone votes yes for a bill, or everyone votes no for a bill. We filtered out the extremes, by only considering bills that had split votes - specifically, bills that had between 171 and 271 yes votes. Unfortunately, this didn't seem to have much of an effect and only reduced the average entropy by perhaps 0.01. We also speculated that the voting may not be strongly along party lines. For example, if 80 Republicans and 180 Democrats vote yes on a bill, the bill is most likely a "Democrat" style bill, but it is difficult to process the fact that the 80 Republicans who voted for the bill were not Democrat. This may cause the neural network to be unable to separate the nodes into different parties.

Additionally, This may be due to the difference in size between the voting relationship graph and the contribution relationship graph. The voting relationship graph had a fraction of the nodes compared to the contribution data, but was also far more dense. In addition, congressmen often have complex voting patterns, and may skip voting on bills which they assume will receive the final vote which they would cast. Finally, not all bills are created equal, and bills proposed which pertain to congressional rules rather than more partisan issues may receive much more bipartisan support. [2] We believe all of these factors contributed to noise in the data set and led to poor results compared to those of the graph composed of contribution data.

Since the voting data did not lead to much signal, the tripartite graph also did not contain much signal. In fact, extending the legislators in the corporate donations bipartite graph to connect to the bills removed all the signal from the corporate donations data as well. Consequently, we do not have any interesting results to report for the tripartite graph.

4.3 KMeans Clustering on Shared Voting Adjacency Matrix

Performing KMeans on Node2vec feature vectors did not produce low entropy clusters. We used the bipartite voting graph to create a square shared voting matrix where cell i,j represents the number of common votes between politician i and j. After constructing this matrix and running KMeans on it, we found low entropy clusters across all congress sessions, each with one predominantly democratic cluster and another predominantly republican cluster. This algorithm was far less complex than Node2vec and SimRank, and much more interpretable, yet we were able to get much better party clusters than either method. We think that this method performed better for the Voting data than Node2vec because

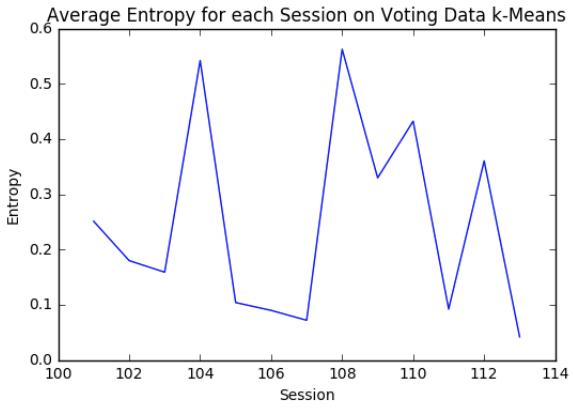

Figure 6: The entropy of the politician party clusters formed using KMeans clustering on Shared Voting Adjacency Matrix across Congress sessions 101 to 113.

this method tailors the feature vector embeddings to focus on political view similarity which facilitates political party cluster creation. In this voting data, Node2vec may have been trying to learn something else in the data that didn't facilitate political party distinction. Figure 3 illustrates the entropy of clusters produced using this algorithm. The results are fairly good with 8 of 12 sessions falling under 0.3 and all sessions below 0.6 entropy. We've also found meaningful anomaly data points where affiliated Republicans have been clusted with Democrats and vice versa. In many of these cases, the anomaly politician usually is found to be a conservative Democrat or liberal Republican.

5. RELATED WORK

University of California San Diego professor James Fowler is at the forefront of combining politics and big data. In the Connecting the Congress: A Study of Cosponsorship Networks paper, James Fowler proposes novel methods of calculating the connectedness between two legislators. The paper first explains the different types of motivation legislators may have for cosponsoring bills from a political science perspective. They first show existing methods to show the connectedness of two legislators. They calculate the shortest cosponsorship distance, or geodesic, between each pair of legislators. They calculate betweenness centrality, which they define as the difference in average length between nodes with a node in the network and with that node removed from the network. They also compute centrality by eigenvector. They then propose a last novel metric that takes into account whether or not the two nodes have mutual cosponsorship and the number of people that cosponsored a given bill. They then use Dijkstra's algorithm to calculate the mean shortest distance to each legislator given these new weights. This paper was easy to understand, and they clearly improve on previous results by incorporating domain-specific knowledge into the algorithm. The paper has stringent rules for connectedness, however, and a more nuanced approach may have picked up more subtle evidence of connectedness. They show that the people who are most connected according to their algorithm are people that seem to be most well connected in real life, such as House majority and minority leaders and committee chairs.

David Lazer, a Professor of Political Science and Computer Science and self-described Network Scientist at Harvard and Northeastern Universities, has also done work in using relationship graph data to study politics. Lazer presents a large obstacle that social scientists have largely been unable to overcome: it's difficult to determine whether individuals seek out friends who have similar political ideologies or whether individuals make friends and then become influenced by their friends and influence their friends and end up with similar political ideologies as their friends. In The Coevolution of Networks and Political Attitudes, Lazer [9] presents four hypotheses. The first is that individuals will tend to have relationships with other individuals with similar political orientations. The second is that in a majority liberal setting, conservatives will tend to be relatively less engaged in the network than liberals. The third is that the political attitudes of people who have ties to each other will tend to become more similar over time, and the fourth is that social influence on political attitudes will be especially powerful among people who are friends, versus among people who work together. Using survey responses from 55 individuals at the beginning, middle, and end the first year of a public policy masters program, and a logistic regression model, Lazer was able to control for individuals' views at the beginning of the year, before any relationships had formed, and conclude that an individual's friends do indeed have a large impact on an individual's political views. This finding was especially powerful given that the data was on individuals who were beginning graduate study of public policy – individuals who wouldn't seem to easily change their minds on politics. Lazer also found that friendship out of class was a much stronger relationship with respect to the probability of influencing political ties than working closely together in class; this is consistent with the notion of "Don't talk about politics at work." There was only insignificant evidence of political homophily – of individuals seeking out friends who shared a political viewpoint with them at the beginning of the year. This isn't to say that this group was heterophilic; individuals were much more likely to make friends with people of the same race or religion. So, the first hypothesis was false (at the beginning of the year), while hypotheses two through four were true.

In Analyzing the U.S. Senate in 2003: Similarities, Clusters, and Blocs [1], the authors introduces various graph analysis techniques into senator voting data. This paper explores: dependencies between pairs of Senators with regards to their voting patterns; the likelihood of senators to vote in a way that is consistent with the chamber outcome; and, inferring vote correlations from membership in ideological groups. While this paper presents interesting results that confirms many characteristics of the Senate, this paper did not introduce any new methodologies but merely applied existing methods in a novel way. Also, the graph analysis methods used in this paper were not very advanced and can be improved upon in our work.

6. FUTURE WORK

In our work so far we have been able to successfully cluster and partition legislators into their respective parties using both company contribution data as well as legislator voting data with a high degree of success. For the contribution data, we were able to form low entropy clusters by running

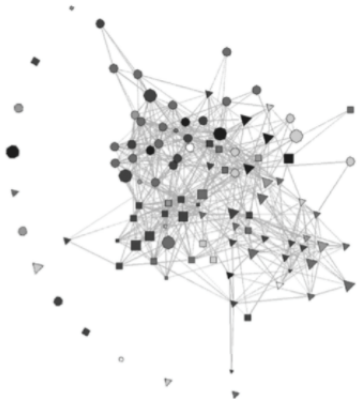

Figure 7: This is the network of the 55 students in Lazer's study. (triangle = Section 1, square/ diamond = Section 2, circle/octagon = Section 3; black = 1 [extremely liberal], white = 6 [conservative] with 4 levels of gray distinguishing intermediate values; largest = became more conservative, smallest = became more liberal, medium size = no change) You can see that many of the conservatives did not become as large of a part of the social network as their liberal counterparts.

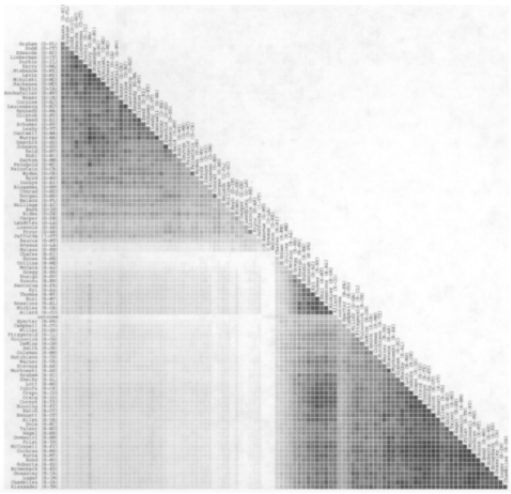

Figure 8: Symmetric dissimilarity matrix graphically illustrating Rajski's distance between all pairs of senators based on their votes in 2003 [1].

Node2vec on a bipartite graph of directed weighted edges from companies to politicians with weighted edges that equal to the amount in U.S. dollars donated. For voting data, we were able to form low entropy clusters by running a straightforward k-Means algorithm on a matrix of the number of votes in common between every pair of legislators.

One of our surprising results has been the poor performance of Node2vec on the voting graph. We showed that a simple feature, the number of votes in common between all pairs of legislators, was capable of separating Democrats from Republicans. Strangely, the neural network was unable to to learn a feature like this and performed poorly on the task. Future work should definitely investigate the shortcomings of Node2vec on this task.

Furthermore, had we been given more time, we would have liked to figure out if there is a way to handle the noise in the voting data and get Node2vec to learn a more useful representation of legislators and bills. Then, we could combine the legislator and bill nodes with the company nodes and explore the similarities between companies and bills. This is definitely another aspect that could be done in the future.

7. CONCLUSIONS

We showed that it is possible to algorithmically determine the political affiliation of a legislator based on his/her corporate donors. This serves as quantitative evidence that legislators are at the very least in some way being paid for holding their beliefs if not actually being paid to maintain a certain belief in Congress. This result is problematic because money should not be influencing politics. There should be further scrutiny of the effects of corporate money in politics.

We also showed that it is possible to algorithmically determine the political affiliation of a legislator based on his/her voting record, which is what we expected. A surprising result, however, was that very simple feature engineering significantly outperformed the Node2vec algorithm which utilizes neural networks. The result suggests that it might be interesting to look into the shortcomings of Node2vec.

In this project, we learned to apply Simrank and Node2vec. We learned that Simrank results in just one feature, the pairwise similarities between nodes, and should be combined with additional features for greater effectiveness. We also learned that Node2vec is a powerful, though difficult to interpret, method for discovering underlying structure in our data. Principal component analysis, clustering, and nearest neighbors can all help interpret the Node2vec embeddings.

8. DIVISION OF WORK

8.1 Danny

I did scraping for the OpenSecrets campaign finance data, scraping different pages to create a roster of which politicians were members of which session of Congress and printing the associated webpage url that contained the donation data from all companies to each Congress member for each session of Congress. Once Charles finished the scraping, I adjusted the donation data for inflation. I wrote the code to then run SimRank on the OpenSecrets campaign finance data. This took many iterations as we tested different strategies to attempt to get the best results. I found outliers in the first principal component and did research to determine what the second principal component was differentiating the Congress

members by.

8.2 Mickey

I implemented generalized structures for storing node and graph data. I wrote several test cases for those structures. I implemented the Simrank algorithm and wrote test cases on small sets of synthetic data for the algorithm. I optimized the Simrank algorithm and reduced the runtime complexity by a factor of $|V|$. I wrote generalized classes and functions for K-Means clustering on node vectors, computing entropy of clusters, and performing PCA and producing scatter plots of the nodes projected onto their principal components.

8.3 Charles

I performed the second part of scraping by using the congressmen IDs and congress cycle years provided by Danny's scraping and fetched the top 100 donations for each of the web pages related to those URLs. I used the scraped contribution data to create a formatted edgelist and node ID map file for the contribution graph. I also wrote a nearest neighbor script for our contribution data. I wrote the script to run K-Means on the contribution and voting SimRank. I generated all Node2vec feature vectors for the two contribution and voting bipartite graph, and also the tripartite graph.

8.4 Jesse

I scraped the data from GovTrack.us relevant to congressman between sessions 100 and 114, which corresponds to years 1987 to 2016. Using this, I formatted the data into a single adjacency matrix for congressmen who have voted on bills together, as well as a complete edgelist of congressmen and the bills which they voted "yes" on. In addition, I also wrote the portion of code for running the K-Means algorithm to cluster nodes based on their SimRank scores. Although it was never used, I wrote code for running KNN on bills and their nearest company neighbors (this was not used because of the poor results received from node2vec on the voting graph). Finally, I wrote the portion of code responsible for running Simrank on the voting graph, as well as many scripts to automate running different portions of the code on all congressional sessions.

9. REFERENCES

[1] A. Jakulin, W. Buntine, T.M.La Pira, and H. Brasher. 2009. Analyzing the U.S. Senate in 2003: Similarities, Clusters, and Blocs. In Political Analysis 17, 3 (2009), 291 − 310. DOI=http://dx.doi.org/10.1093/pan/mpp006

[2] @. OpenSecrets. Retrieved December 10, 2016 from http://www.opensecrets.org/

[3] Anon. GovTrack.us. Retrieved December 10, 2016 from https://www.govtrack.us/

[4] Glen Jeh and Jennifer Widom. 2002. SimRank: a measure of structural-context similarity. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02). ACM, New York, NY, USA, 538 − 543. DOI=http://dx.doi.org/10.1145/775047.775126

[5] Antonellis, I., Garcia-Molina, H., and Chang, C.-C. Simrank++: Query rewriting through link analysis of the click graph. In Proceedings of VLDB (Dec 2008), pp. 408 − 421.

[6] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

[7] Anon. Sanford Bishop. Retrieved December 10, 2016 from https://en.wikipedia.org/wiki/Sanford_Bishop

[8] Anon. Citizens United v. FEC. Retrieved December 10, 2016 from https://en.wikipedia.org/wiki/Citizens_United_v._FEC

[9] David Lazer, Brian Rubineau, Carol Chetkovich, Nancy Katz, and Michael Neblo. 2010. The Coevolution of Networks and Political Attitudes. Retrieved December 10, 2016 from http://www.lazerlab.net/sites/default/files/publications/-The%20coevolution%20of%20networks%20and%20political-%20attitudes.pdf

[10] Jennifer Steinhauer and Derek Willis. 2011. Congressional Voting Records Show Few With Perfect Attendance. (2011). Retrieved December 12, 2016 from http://www.nytimes.com/2011/11/01/us/politics/congress-voting-records-show-few-with-perfect-attendance.html