Conflicts of Interest?
The Interplay of Money and Politics in the United States House of Representatives

Brian Hicks (bhicks2) and Michael Johnson (mikejohn)

December 12, 2016

Abstract

In this paper, we present a network-based analysis on the relationship between the voting behavior of sitting politicians in the House of Representatives of the 113th Congress and the campaign donors who funded their elections. Our investigation consisted of composing a network using open source data connecting politicians based upon similar voting records and similar sources of donors. We then leverage two types of analysis based on, (1) least-squares regression, and (2) community detection, to provide insight into whether or not making donations to a politician grant a level of influence (if any) over how the politician votes. Through this analysis, we found that a relationship does exist between how politicians vote with respect to similar donation sources, but it seems less dependent on the amount of money involved in the donations and more significantly linked with the act of donating at all, meaning politicians are more likely to vote together if they have a higher number of similar donors.

1 Introduction

Government and law in the United States of America are not controlled directly by the will of the populace. For better or worse, the concrete decisions of governance are not made in the poll-booth, but rather in the chambers of the Capitol, where 535 elected representatives execute the powers of the legislative branch. However, the process of running for election to these positions is exceedingly expensive, which leads candidates for political office to rely heavily on large financial contributions from donors to fund their campaigns. Afterward, at least according to common belief, these benefactors have some measure of influence over the recipient, and can thus push them toward decisions that serve a personal or private interest, rather than a public one.

Although the question of what role, if any, money should play in campaigns for political office and politics in general is a polemic one, our paper does not seek to vindicate or refute any particular position on this issue. Instead, our goal in this investigation is to better understand *how* the interplay between donations and a politician's behavior manifests — information that is essential to any form of educated debate on the issue. To this end, our project consists of two main parts: data collection and data analysis, both of which are described in greater detail below. For the purposes of this investigation, we have elected to focus our analysis on members of the House of Representatives in the 113th Congress, whose members held office between 2013 and 2015.

1.1 Data Collection

In order to analyze and interpret the interplay between financial contributions and political behavior, it is necessary to have access to data that describes such relationships. On the surface, this seems an easy task: both how a politician votes and the gifts and donations that they receive are all a matter of public record. Unfortunately, however, existing data sets do not fulfill the needs of this investigation. More specifically, no single existing set of data connects a politician's actual voting record with the donations that they received on the campaign trail. Thus, a not insignificant portion of our project revolves around the assembling of the disparate data sets into a unified whole.

1.2 Network Analysis

Although the compilation of a complete data set as described above is an essential part of our project, it is not the main focus of our paper, but rather an indispensable first step. Instead, the main thrust of our project centers on the analysis of this political-financial network to better understand the relationship between money and behavior in the legislative process, with a particular emphasis on network-based models and analyses. More specifically, over the course of this project, we hope to leverage network-analytic algorithms and tools to develop a deeper understanding of the latent information hidden within the network, particularly with regard to interpersonal structures, such as sub-communities within the congressional chambers. The following software packages were used in the analysis of the data: scikit-learn [1], SciPy [2], NumPy [3], and the Stanford Network Analysis Program (SNAP) [4].

2 Related Work

There is little literature on a network based approach to understanding the interplay between money and politics that we plan to pursue in this project. As a result, we will discuss prior work on the type of algorithms and measurements we will find necessary to provide useful insight into the nature of the relationships.

2.1 Community Structure in Social and Biological Networks (Girvan & Newman, 2002)

The property of community structure, in which network nodes are joined together in dense groups, can provide useful information about the structure of the network. The main focus of Girvan and Newman [5]'s paper is the formulation of a new algorithm for the identification of underlying community structures in network data. Eschewing the traditional hierarchical approaches, they instead advocate for a divisive approach, wherein communities are identified primarily by the boundaries between them, rather then the most central members. To this end, they expand Freeman's definition of betweenness (usually applied to nodes) to edges, by defining the *edge-betweenness* of a given edge as the number of shortest paths that run along it. Their method provides insight into which edges are the "bridge" between communities, and, as a result, identify informative community divisions within the network.

In their analysis of the efficacy of their proposal, Girvan and Newman show that, given a network with already known community structures (e.g., Division I college football), their algorithm identifies the communities with high accuracy. However, there is some evidence (from their tests on food webs) that the algorithm performs best on sparse networks, and cannot reliably identify the communities in a dense network [1]. Moreover, the paper recognizes the limitations of the algorithm due to its speed and lack of generality to weighted and directed graphs. The algorithm runs in the worst-case $O(m^2n)$ time where m is the number of edges in the graph and n is the number of vertices, and $O(n^3)$ for sparse graphs, meaning the run time of this algorithm will likely be intractable for large or dense networks.

2.2 Finding and Evaluating Community Structure in Networks (Newman & Girvan, 2004)

In their second paper [6], Newman and Girvan further develop their divisive algorithm for finding community structures in a network. They begin by proposing an alternate metric for a betweenness score, defined as the expected number of times an edge will be crossed during a random walk between two nodes, which allows for considerations of networks where information may not travel only by the most direct route. Furthermore, the algorithm now takes an iterative approach to determining community structure by gradually removing edges from the network based on the measure of betweenness, splitting the graph into communities, and recalculating the betweenness scores of the edges after each

[1] Girvan and Newman note however, that the dense networks used in testing for [5], were food-webs, which do not have the *a priori* community divisions that are available in other data sets. As a result, the inefficacy in such circumstances may be a result of the absence of clearly delineated communities, and not a fundamental failure of the algorithm

removal. Lastly, the paper proposes an objective measure for choosing the number of communities in the network, which they call the modularity.

Again, the authors recognize the computational demands of their algorithm, which runs in $O(n^3)$ for sparse graphs, limiting its use, although as noted in the paper, this could be mitigated by parallelizing the calculations of betweenness. The paper also points out that betweenness scores can be calculated for edges of directed networks by only considering paths that follow edges in the forward direction.

2.3 Finding community structure in very large networks (Clauset, Newman, & Moore, 2004)

The algorithms for the discovery and analysis of community structure in networks so far are limited by computational demands. This paper presents a hierarchical agglomeration algorithm for detecting community structure by greedily optimizing the modularity, a measure for the quality of the divisions of the network into communities [7]. The run time of this algorithm is $O(md \log n)$ where n is the number of nodes, m is the number of edges, and d is the depth of the dendogram describing the community structure.

The algorithm begins from a state where each node is its own community and repeatedly joins two communities depending on the link which optimizes the increase in modularity. The authors demonstrate the algorithm on a large network of co-purchasing data from the online retailer Amazon.com, which discovers the existence of communities within the network, corresponding to specific topics or genres of books or music. This greedy approach to optimizing the resulting modularity of joining two communities offers considerably improvements in speed to previous algorithms discussed.

3 Model and Analysis

3.1 Data

As we described in the introduction, there is no existing data set that links together the financial gifts and donations that politicians receive with their voting behavior once in office. Thus, a considerable portion of our efforts prior to the investigation was devoted to combining existing data into a cohesive whole that we could use to generate the networks that could be used in our analysis. In the end, we drew on data provided by two reputable, non-partisan political watchdogs: OpenSecrets.org [8] and GovTrack.us [9]. The first is a research group which tracks money in American politics, as well as its effects on the public, by amassing data about campaign finance and lobbying. Meanwhile, the latter is a project of Civic Impulse, LLC. which records and makes available the voting results of every bill and motion that is considered in either of the congressional chambers. Both of the data sets that we used are made freely available to the public.

For the purposes of this analysis, we restricted our focus to the members of the House of Representatives during the 113th Congress, who held office from January of 2013 through January of 2015. In limiting the scope of the investigation to only the House, and not the Senate, we are able to avoid the added complexities of the Senate's staggered elections (i.e., approximately one-third of the Senate is up for re-election at any given time, which could produce unexpected or misleading inter-node links as a result of the constant influx of new, freshman senators over the course of a single term).

3.2 Networks

In order to fully understand the relationship between money and a politician's behavior, we used the data compiled above to generate three distinct networks. The first network explores the relationships between politicians with regard to how they cast their votes. To do so, we first define the following similarity metric S_{vote}:

Let Y_i be the set of all bills and motions on which politician i voted *Yea*. Similarly, let N_i be the set of all bills and motions on which i voted *Nay*. Lastly, let $V_i = Y_i \cup N_i$ (i.e., V_i is the set of all bills and motions for which i submitted a vote, regardless of whether it was *Yea* or *Nay*). We then define

the vote-similarity score for an arbitrary pair of politicians (i, j) as

$$S_{vote}(i,j) = \frac{|Y_i \cap Y_j| + |N_i \cap N_j|}{|V_i \cap V_j|}$$

Intuitively, the value of $S_{vote}(i, j)$ can be seen as the number of bills that i and j voted similarly on (i.e., both *Yea* or both *Nay*) as a fraction of the total number of bills that they both voted on, and therefore, by extension, the probability of a given pair of Representatives agreeing on a bill. As a metric of similarity, S_{votes} provides great insight into how similar the ideologies of two politicians are, as we would expect that politicians who both submit the same vote on a bill believe similarly about its ideological merits.

With this metric of similarity, we then construct a vote-similarity network. This is an undirected, weighted network in which each node represents a member of the House of Representatives. An edge between two nodes i and j represents that some level of ideological similarity (as measured by the vote-similarity metric) exists between those two politicians, and the weight of this edge is given as $S_{vote}(i, j)$. Ignoring weight zero edges, there are 100,566 edges in the graph, which, with 449 nodes[2], makes this graph extremely dense — only 10 edges short of the maximum.

Our second graph seeks to represent the the similarity of a given pair of politician's financial backing. In other words, we seek to quantify how much of their campaign donations came from the same organization. To do so, we define the money-similarity metric S_{money}:

Let D_i be the set of groups that donated to a politician i. Let the function $\delta_i : D_i \to \mathbb{R}$ be a mapping from a donor d to the amount of money they donated to politician i. We then define the similarity metric S_{money} for a given pair of politicians (i, j) as

$$S_{money}(i,j) = \frac{\sum_{d \in (D_i \cap D_j)} \delta_i(d) + \delta_j(d)}{\sum_{d \in (D_i \cup D_j)} \delta_i(d) + \delta_j(d)}$$

Intuitively, this metric gives the amount of money donated to i and j from shared backers as a fraction of the total amount of money anyone donated to i and j.

Given this similarity metric, we then can construct our second network: the money-similarity network. Just as in the vote-similarity network, this graph is an undirected, weighted network for which each node represents a Representative. However, in this graph, the edges are not a measure of ideological similarity, but rather of financial backing similarity. Thus an edge between two nodes i and j means that there is some level of similarity to the two politicians' financial backings. The weight of this edge, given as $S_{money}(i, j)$ then gives the strength of this similarity. Ignoring 0-weight edges, there are 100,566 edges in the graph (just as in the vote-similarity graph). With 449 nodes, this means that the graph must be extremely dense.

Finally, our last graph is intended to explore the similarity of a given pair of politician's donors in a similar manner to the graph described above. For the purposes of this graph, however, the amount of money received from shared donors is no longer relevant, just the count of shared donors. Thus we define a similarity metric S_{donor} to quantify this relationship:

As before, let D_i be the set of donors that made a contribution to politician i. We then define the similarity score between two politicians i and j as

$$S_{donor}(i,j) = \frac{|D_i \cap D_j|}{|D_i \cup D_j||}$$

Ultimately this metric is just the Jaccard index of set similarity. Intuitively, it gives us a measure of the number of shared donors as a fraction of the total number of donors to a given pair of politicians i and j. In many respects, it is a reformulation of S_{money}, as it also attempts to provide information about the similarity of the financial backing of politicians, but it is also unique in that all donors are treated equally, regardless of the amount of money they donated.

[2]Although there are only 435 voting members of the House of Representatives, our data also includes non-voting members such as the Representative from Guam, Madeleine Bordallo, as well as those who replaced other Representatives that did not complete their term. This is why we have 449 nodes, as opposed to the expected 435

With this metric in hand, we can then construct our third and final network: the donor-similarity network. As with the other two networks, the donor-similarity network is an undirected weighted graph where each node represents a politician. For the purposes of this graph however, an edge between two nodes i and j indicates that there is some amount of overlap in the list of groups that donated to the two politicians. The weight of this edge, given by $S_{donor}(i, j)$, can then be interpreted as how strong this similarity is. With 100,566 edges between 449 nodes, this network, like the others, is extremely dense — almost the point of being complete.

3.3 Analysis

The central question of our investigation is what, if any, is the relationship between money and politics. In a very broad sense, there are two general hypotheses we could form in response to this question:

(I) Making donations to a politician grants a level of influence over how that politician votes (and thus the laws that pass)

(II) Making donations to a politician does NOT bestow a significant level of influence over how that politician votes (null hypothesis).

Both of these hypotheses make different predictions about what we would see in the data. In the first case, with Hypothesis I we would expect to find that the donor- and money-based metrics would be good predictors of the voting behavior of the members of the House of Representatives. In other words, given information about a politician's financial backing, we should be able to make reasonably accurate predictions about how that politician will vote, if Hypothesis I is true.

Hypothesis II on the other hand, makes different predictions. Under this hypothesis, we would expect to find that the connection between money and politics is weak at best, which would manifest itself as an inability to accurately predict voting behavior based on information about a politician's financial backers.

Given the networks we have extracted from our data and the information contained therein, both of these hypotheses are easily testable. In order to do so, we will attempt to predict the voting behavior of politicians (already known *a priori* through GovTrack.us's data [9]), using the financial information we gathered from OpenSecrets.org [8]. We intend to leverage two types of analysis (least-squares regression and community detection), each of which is described in further detail below, both in terms of the algorithm as well as the specific application.

3.3.1 Least Squares Regression

Under Hypothesis I, as described above, knowledge about a politician's financial background should give us a relatively accurate prediction about how that politician will vote. This would suggest that, if Hypothesis I is true, we should be able to predict $S_{vote}(i, j)$ for a given pair of politicians (i, j) given $S_{money}(i, j)$ and/or $S_{donor}(i, j)$ — in other words, knowing how similar two politicians' financial backing is should give us insight into how similarly they vote.

In order to test the correctness of this prediction, we ran least-squares linear regression on our data, attempting to predict $S_{vote}(i, j)$ as a function of $S_{donor}(i, j)$ or $S_{money}(i, j)$. The intuition behind the analysis is relatively simple: given a collection of n data points $x_1, \ldots, x_n$ and their mappings $y_1, \ldots, y_n$, we want to find a value of m and b such that we can minimize the objective function

$$\sum_{i=1}^{n} (mx_i + b - y_i)^2$$

For the purposes of our analysis, however, the actual values of m and b are not of particular importance. What is actually of interest is how accurate the prediction really is — i.e., how well does our regression function fit the data. One of the standard metrics for this is the coefficient of determination, written as R^2. Essentially, this value gives a sense of how much of the variation in our the actual data can be explained by a our fitted function. R^2 ranges from 0 to 1, with 1 being a perfect fit. Thus, in interpreting the following regressions, we can compare them based on the value of R^2.

Running the analysis described above on the 100,566 pairs of politicians in our data gives us the results shown in Table 1.

Table 1: Least Squares Linear Regression Results

	S_{money}	S_{donor}
R^2	0.1716	0.3097
p-Value	< 0.05	< 0.05

We can see from the p-Values that both of the regression lines are significantly better at explaining the variation in our data than the null-model (i.e., $m = 0$). However, it is also true that S_{donor} manages to explain much more of the variation seen in our data – in fact it is nearly twice as effective. Based on these results, we can conclude that both S_{money} and S_{donor} have statistically significant predictive power with respect to S_{vote}. Neither is particularly accurate, as at most they explain 30% of the variation in S_{vote}, but of the two S_{donor} can account for almost twice the amount of variation in the data as S_{donor}. At a high level, it appears that S_{donor} is likely to provide a more accurate estimate of the value of S_{vote} than S_{money}. The significance of these results will be discussed below in Section 4.

3.3.2 Community Detection

As we've alluded to above, one of the most intuitive ways to categorize and/or analyze a politician's voting behavior is by comparing it to others'. Our definition of S_{vote}, and thus our least squares regression from the previous section, follows directly from this intuition. We can build further upon this intuition to gain a broader understanding of how members of the House of Representatives vote with respect to one another. Specifically, we can explore what, if any, voting blocs exist within the House, and how we can leverage this information in order to gain insights into the relationship between money and politics.

Community detection is a very well researched problem within the field of computer science and network analysis. Many different algorithms exist that attempt to find these sub-structures in a larger network. The two algorithms that most interest us for this investigation are the Girvan-Newman algorithm described in [5] and [6], and the Clauset-Newman-Moore algorithm from [7]; also, briefly, introduced in the "Related Work" section of this paper.

In our initial attempts to explore this question of voting blocs, we favored the Girvan-Newman algorithm for the analysis. A divisive clustering algorithm, the procedure was relatively straightforward:

1. Calculate the betweenness score for each edge in the graph (i.e., how many shortest-paths pass along a given edge)
2. Remove the edge(s) with the highest betweenness from the network
3. Repeat steps 1 and 2 until all nodes are in their own, single-node communities.

For the purposes of analyzing the vote-similarity graph, we formulated a new definition of "shortest-path" in order to take into account the weights of our edges (S_{vote}, which represents, more or less, the probability that a given action propagates from one node to another). For a path that passes along edges e_1 through e_k, the "length" of the path would be given by

$$\prod_{i=1}^{k} (1 - S_{vote}(e_1))$$

The intuition behind this path-length metric was that the ideal path to take between two nodes was the one that minimized the probability of an action being lost (i.e., not propagating from one node to the next).

Although this was by far our preferred method of analysis, as the intuition behind it was strong, the computational resources required made the problem intractable. One major problem was that, under our definition of shortest path, the fact that our edge weights were all in the range [0, 1] (since they are probabilities) made them into the multiplicative equivalent of negative-weight edges, and thus our network was filled with negative cycles, meaning there was no tractable algorithm to find the shortest path.

Even if our definition of the shortest-path had been feasible, however, the Girvan-Newman algorithm was far too slow for our problem to be tractable. Even though the number of nodes in our graph is relatively small (with only 449 nodes), it is incredibly dense, to the point that it is nearly a complete graph. As a result, the edge-betweenness calculations were too slow to be able to finish in a reasonable amount of time. Ultimately, we were forced to abandon the Girvan-Newman algorithm in favor of the Clauset-Newman-Moore algorithm.

Described in [7], the CNM algorithm is a much faster computation for identifying communities. The algorithm is based upon a modularity score

$$Q = \sum_i \left(e_{ii} - a_i^2\right)$$

where e_{ij} is the fraction of edges that connect a community i to another community j, and a_i is the proportion of edge-ends that connect to a node in community i. Intuitively, this measure of modularity is a measure of how unlikely a given arrangement of edges is to have arisen by chance in a random graph — i.e., how "surprised" would we be if we found this set of edges in a randomly generated graph.

With this score Q in mind, the algorithm then follows a very simple agglomerative process: at each step of the algorithm, greedily combine the communities to maximize the value of Q. The community structure is then given by the community divisions at the point when Q is the highest.

To run this new CNM algorithm, we elected to transform our vote-similarity graph in order to avoid the issue of edge weights. The transformation process was relatively simple. We began by calculating the average weight of the edges $\overline{S}_{votes}$. Intuitively, this value is the standard level of agreement between politicians. We then classified any edges whose weight was less than $\overline{S}_{votes}$ as negative edges (the intuition being that on anything controversial, these pairs were unlikely to agree) and all of the edges with a weight greater than $\overline{S}_{votes}$ as a positive edge.

Next, because we recognized that behavior was unlikely to propagate down a negative edge, we removed all of the negative edges from the graph. With our graph thus transformed, we then ran the CNM algorithm to identify the voting blocs in the network.

To those familiar with the highly polarized state of affairs in the United States Legislature, the results of our clustering algorithm will be unsurprising. Specifically, the algorithm identified two primary clusters[3] with a modularity score of 0.4108, suggesting reasonably strong community structuring. The two largest communities were, unsurprisingly, split entirely by party lines (i.e., one community was 100% Republican and the other was 100% Democrat). While this is not a surprising result, it provides a useful sanity check to verify that the clustering algorithm works properly.

Although this background is necessary to understand process by which we arrive at the clusters, the communities in the vote-similarity network are not our primary concern here. In reality, our goal is to identify the relationship between money and politics, which seek to do by analyzing the predictive power of the S_{money} and S_{donor} metrics and networks.

In this case, if Hypothesis I is true, we should expect to see that the money-similarity and donor-similarity graphs, when run through the same process described above, produce voting blocs very similar to those found when analyzing the vote-similarity network. And, to some extent, this is exactly what we find. Specifically, in both of those networks the CNM algorithm identifies two large communities that comprise the majority of the nodes, divided largely along party lines. However, it is important to note that the money-similarity graph's communities have a modularity of only 0.164, while the donor-similarity graph has a modularity of 0.263. The latter's is not particularly bad, but the money-similarity graph's community structure appears to be very weak — and neither's is as strong as the vote-similarity network.

[3]There were a total of 13 communities identified, but the largest two contained 97% of the nodes

With this caveat in mind, we can now compare the two main communities identified by the CNM algorithm, analyzing them separately as the Republican-Majority Community and the Democrat-Majority Community, as shown in Table 2 and Table 3, respectively. Please note that the three bottom rows labeled "Jaccard" refer to the Jaccard index, which is a similarity score for sets. It is defined, for an arbitrary set A and B as

$$\frac{|A \cap B|}{|A \cup B|}$$

.

Table 2: Republican-Majority Community

	Vote	Money	Donor
Democrats	0%	5.8%	1.8%
Republicans	100%	94.2%	98.2%
Jaccard (Vote)	N/A	0.712	0.914
Jaccard (Money)	0.712	N/A	0.739
Jaccard (Donor)	0.914	0.739	N/A

Table 3: Democrat-Majority Community

	Vote	Money	Donor
Democrats	100%	78.7%	96.1%
Republicans	0%	21.3%	3.9%
Jaccard (Vote)	N/A	0.739	0.915
Jaccard (Money)	0.739	N/A	0.747
Jaccard (Donor)	0.915	0.747	N/A

In looking at the results of the CNM community-detection algorithm, we see that of the two finance-based metrics, the donor-similarity graph performed much better than the money-similarity network. The Jaccard Indices between the vote- and donor-network-based communities were 0.914 and 0.915 — much higher than the 0.712 and 0.739 that the money-similarity graph achieved. Thus, it appears that in terms of predicting the voting blocs, the donor-based network performs much better than the money-based one. The significance of these findings is explained in Section 4.

4 Results and Discussion

Recall from the beginning of Section 3 that our analysis above was intended to provide evidence to support one of two hypotheses: Hypothesis I, which asserts that there *is* a correlation between money and political behavior and Hypothesis II (the null hypothesis), which posits that there is not. We further explained that if Hypothesis I is true, we should expect to find that finance-based similarity metrics should be able to accurately predict voting behavior of politicians, and that an absence of such trends would support Hypothesis II.

Based on the results of the least-squares linear regression and the CNM community detection algorithm, it appears that there is some merit to Hypothesis I. Specifically, we found in both cases that the money-similarity metric and the donor-similarity metric had at least mediocre ability to accurately predict the voting behavior. In the case of the linear regression analysis, we found that both the S_{money} and S_{donor} models are statistically significant with respect to the null model, and we say in the community detection algorithm that even in the worst case (the Republican-Majority

Community) there was a Jaccard Index of 0.712, which is relatively high (although not spectacular). All of these results support, to some extent, the hypothesis that there *is* a relationship between the donations that a politician receives and their behavior once elected.

Although the findings above are accurate at a high level, there are a few important caveats that should be noted when interpreting the data. The first issue to address here is that the two metrics S_{money} and S_{donor} are not equally powerful predictors. That is to say, they do not attain similar levels of accuracy in their predictions of the voting behavior. This was apparent in every level of our analysis. In the linear regression test, we found that the R^2 (a measure of the goodness of fit) value for the S_{money} model was almost half of S_{donor}'s R^2 — indicating one-half of the explanatory power. Similarly, in the community detection analysis, the modularity score for the money-based network was very low at only 0.164. Such a low modularity score is indicative of a fairly weak community structure, which means that although the absolute accuracy of its clusters was not abyssal, they were not clearly defined, suggesting a high level of uncertainty in the predictions. In comparison, the S_{donor} model had a a comparatively high modularity score at 0.263, which suggests, in combination with the high Jaccard index value with the S_{vote} model, suggests that it has (a) a high rate of accuracy and (b) is reasonably certain about the divisions — both of which are indicative of a relatively good predictor, and definitely better than S_{money}.

Another issue that should be addressed is the fact that there are certainly more factors at play in the voting behavior than just money. To see this, consider the S_{vote} communities identified by the CNM algorithm — they are divided strictly by party allegiance, without a single defector. This suggests that there is some element of political party dictating voting behavior as well, even without taking into account financial incentives to vote one way or the other.

With these caveats in mind, we come back to our original question: What is the relationship between money and politics? Based on our analysis, it appears that Hypothesis I is correct — there is a connection between the donations a politician receives and the way that they vote once in office. However, somewhat counter-intuitively, it appears that it is *not* the amount of money contributed that is important, but merely the act of donating. This conclusion arises from the fact that S_{donor} provides a much more reliable prediction of voting behavior than S_{money}, which suggests that *who* your donors are is more important than *what* they give you. As a caveat, we do recognize that it is possible that this is a correlative relationship, and not causal. That is to say, it is possible that politicians receive donations *because* of their ideology and voting behavior, rather than voting a certain way because of the donations.

The relationship between money and politics is a hotly debated issue. Some feel that the ability to donate to your preferred candidate is a form of expression protected by the First Amendment, while others believe that it creates substantial risk of corruption and cronyism. While this paper does not attempt to address the issues that underlie these beliefs, we have sought to address whether this is a correlation between money and political behavior even exists. Ultimately, we have concluded that there *is* a relationship between these two — how this ultimately affects the debate remains to be seen.

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[2] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001.

[3] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[4] J. Leskovec and R. Sosič, "Snap: A general-purpose network analysis and graph-mining library," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 1, p. 1, 2016.

[5] M. Girvan and M. Newman, "Community structure in social and biological networks," in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 8271–8276, 2002.

[6] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, 2004.

[7] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, p. 066111, Dec 2004.

[8] The Center for Responsive Politics (Washington D.C.), "Opensecrets.org: The web site of the center for responsive politics." https://www.opensecrets.org, 1990.

[9] Civic Impulse, LLC, "GovTrack.us." https://www.govtrack.us, 2004.

Parter Contributions

- Brian Hicks: wrote Section 1, 3, and 4; formulated the plans for analysis (e.g., which algorithms, what similarity metrics); wrote the code for linear regression using scipy's linear regression; wrote the code for CNM clustering using snap's CNM function; interpreted the results.
- Michael Johnson: wrote Section 2; coded up most of the non-snap algorithms, including some that were not used, such as attractiveness-based community detection; wrote code to pull together a cohesive data set from a wide variety of disparate data sets and adapt it to changing demands; proofreading.
- Overall: we feel that the division of labor was fair and equitable.