Recovering Social Networks by Observing Votes

Benjamin Fish and **Yi Huang** and **Lev Reyzin**

Department of Mathematics, Statistics, & Computer Science
University of Illinois at Chicago, Chicago, IL 60607
{bfish3,yihuang89,lreyzin}@uic.edu

Abstract

We investigate how to reconstruct social networks from voting data. In particular, given a voting model that considers social network structure, we aim to find the network that best explains the agents' votes. We study two plausible voting models, one edge-centric and the other vertex-centric.

For these models, we give algorithms and lower bounds, characterizing cases where network recovery is possible and where it is computationally difficult. We also test our algorithms on United States Senate data.

Despite the similarity of the two models, we show that their respective network recovery problems differ in complexity and involve distinct algorithmic challenges. Moreover, the networks produced when working under these models can also differ significantly. These results indicate that great care should be exercised when choosing a voting model for network recovery tasks.

Introduction

One approach to investigating voting data assumes that agents' votes are independent of one another, conditioned on some underlying (sometimes probabilistic) model of ground truth. This is usually an unrealistic assumption, leading to a more recent line of inquiry which asks how the social network structure of the voters affects the relationship between votes. Each agent in a social network expresses a position (votes for or against a bill, prefers one brand over another, etc.) that is influenced by their social connections. In this view, it is possible to detect the organization and evolution of voting communities by looking at the social network structure. The literature on congressional and political voting networks focuses on detecting community structure, partisanship, and evolutionary dynamics [1, 7, 9, 13, 17, 18], while the literature on idea propagation investigates how to best maximize the spread of ideas [3, 8].

However, it is often not necessarily clear how to build this social network graph. For example, Macon et al. give a few different variants on how to define the social network of the voters of the United Nations [9]. In this approach, different graphs may reveal different aspects of the social network structure.

This corresponds neatly with a typical view in social choice theory that votes are manifestations of subjective preferences. At the other extreme, a voter votes according to a noisy estimate of the ground-truth qualities of the possible choices on which he or she is voting. While both are over-simplified extremes, it is useful to consider the extremes in order to investigate their consequences [4]. In this paper, we take the latter approach.

This is the maximum likelihood approach to voting. In this approach, a fixed probabilistic model is used to determine the relationship between how the ground-truth qualities of the possible choices lead to the noisy estimates of those qualities, which the voters then use to vote. This probabilistic model takes into account the social network structure in which the voters are embedded.

In this approach, it is typically assumed that the social network of the voters is known. The goal is then to find the correct choice from votes, as tackled by Conitzer and then others [4, 14, 16]. This can be made more difficult depending on the structure of a social network, which may enforce the wrong choice by aggregating individual opinions over dense subgraphs, leading voters with low degree to possibly change their mind to the majority view of the subgraph.

In practice, however, the social network is not known and it is not necessarily clear how to infer the graph. In this paper, we tackle the problem of inferring the social network from the votes alone. We discuss two similar but distinct voting models in the vein of Conitzer [4], and show how to recover the graph given the votes under these voting models and under several notions of what it means to recover the graph. We show that your ability to learn the graph from the votes is highly dependent on the underlying voting model - in some settings, it is computationally hard to do so but not in others. Moreover, we demonstrate that the resulting learned graphs can differ significantly depending on which underlying voting model is assumed.

Models and results

We give results for two similar models: an edge-centric model, which Conitzer calls the *independent conversation model* [4], and a vertex-centric model, introduced in this paper, which we will call the *common neighbor model*. Similar to some existing models, the common neighbor model is, for instance, equivalent to the "deterministic binary majority process" run for one step (where the initial assignment is random). This process was examined by Goles and Olivos [6] and related work, e.g. by Frischknecht et al. [5], and it has been used in the press to illustrate the disproportionate influence of certain voters [15].

In both models, there is an unknown simple undirected graph G on n vertices. Each vertex is an agent, who can vote "-1" or "1". Both models describe how each agent votes in one round of voting. We consider m rounds of voting and in each round every vertex votes, leading to a sequence of vote sets $V^{[m]} = V_1, \dots, V_m$, where each V_i is the set of votes from all voters. The problem is to recover G from $V^{[m]}$.

First, we define the independent conversation model, a two-step process where edges represent conversations between voting agents, and each agent votes according to the majority outcome of his conversations.

Definition (**independent conversation model**). *First, each edge flips a coin i.i.d. that with probability p is 1 and with probability $(1-p)$ is -1. Then each vertex votes according to the majority of its adjacent edges' preferences. If there is a tie, then it is broken in favor of voting 1 with probability q and -1 with probability $1-q$.*

This process is depicted for a particular graph in Figure 1. Note that the set of votes V_i only includes the final votes, not the initial preferences.

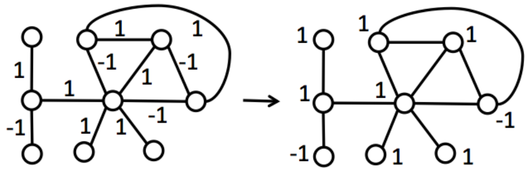


Figure 1: Left: the outcome of pairwise "conversations" between connected neighbors. Right: the resulting votes. For simplicity, the edge probabilities are not depicted.

The common neighbor model is similar, except here the initial preferences are on the vertices, not the edges:

Definition (**common neighbor model**). *Each vertex initially flips a coin i.i.d. that with probability p is 1 and with probability $1-p$ is -1. Then each vertex votes 1 if more adjacent vertices' initial preferences were 1 then -1 and vice versa. If there is a tie, then it is broken in favor of voting 1 with some probability q and -1 with probability $1-q$.*

This process is illustrated in Figure 2.

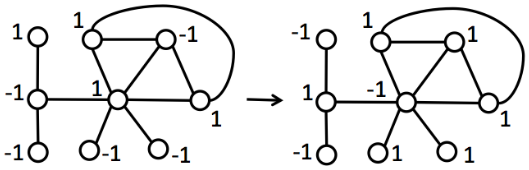


Figure 2: Left: the initial preferences of the nodes. Right: the resulting votes. For simplicity, the preference probabilities are not depicted.

It is straightforward to see how they are different. In the independent conversation model, two vertices' votes are independent of each other if and only if they do not share an edge, while in the common neighbor model, they are independent if and only if they have no common neighbors.

Our main contribution consists of algorithms to recover the hidden graph G from the votes, lower bounds, and experiments for both models.

Our results span a few different notions of what it means to recover the unknown graph. First, we ask whether there exists a polynomial time algorithm (using only a polynomial number of votes in the number of voters) that finds the unknown graph G when the votes were drawn from G. We show the following:

Result 1. *In the independent conversation model, there is an algorithm that, given a number of votes and time polynomial in the number of vertices, recovers the unknown graph when $p = 1/2$ with high probability. Moreover, for constant $p \neq 1/2$, an exponential number of votes are required to recover the graph. (See **Observation 1** and **Theorem 2**.)*

Our algorithm is a statistical test for edges between pairs of vertices by calculating sample covariances of their votes, which here measures how likely it is that two voters vote the same way. This is very similar to how voting networks are often constructed in the literature [2, 9, 17]. This result can then be seen as a formal motivation for why this type of method should be used.

Result 2. *In the common neighbor model, no algorithm can exactly recover the unknown graph. (See **Observation 7**.)*

The above two results motivate us to consider other notions of what it means to recover the graph because the graph is generally not recoverable efficiently here. Moreover, in a setting where there is not necessarily a ground-truth graph from which the votes were drawn, we are still interested in finding a graph that explains the votes. We make this precise by asking for the maximum likelihood estimator (MLE) graph, that graph that maximizes the probability of the observed votes over the distribution of votes induced by a fixed voting model. As is standard for the maximum likelihood estimator, we assume that the prior over graphs is uniform.

Result 3. *For the independent conversation model, there is a polynomial-time bounded-probability random reduction from a #P-complete problem to finding the likelihood of the MLE graph. (See **Theorem 5**.)*

This result is an indication that computing the likelihood of the MLE graph is difficult, since if there were an efficient algorithm to compute this quantity then there would be an efficient algorithm to solve a #P-complete problem that succeeds with high probability.

In the common neighbor model, we investigate a third approach to finding a graph that explains the votes. Given that we recover graphs in the independent conversation model using covariances between votes, it is natural to ask whether it is possible to find a graph whose expected covariances are close to the observed covariances in the common neighbor model. We show that even if you were to know the expected covariances exactly, it would still be computationally difficult to find such a graph.

Result 4. *For the common neighbor model, finding a graph with given expected covariances between votes is at least* *as hard as recovering an adjacency matrix from its square. Moreover, a generalization of this problem, namely the generalized squared adjacency problem, is NP-hard. (See **Observation 8** and **Theorem 10**.)*

The squared adjacency problem and its generalized version are defined as follows:

Definition. *The input for the squared adjacency matrix problem is a matrix B, and the decision problem asks if there is an adjacency matrix A of a simple graph such that $A^2 = B$.*

Definition. *The input for the generalized squared adjacency problem is a collection of n^2 sets S_{ij}, and the decision problem asks if there is a simple graph whose adjacency matrix is A such that $A^2_{ij} \in S_{ij}$ for each entry A^2_{ij} of A^2.*

To the best of our knowledge, the squared adjacency matrix problem is not known to be NP-hard nor is it known to be in P. It is a difficult open problem in its own right, and other versions of it have been proven NP-hard [10]. It is equivalent to the special case of the generalized version where the set sizes are exactly one.

The independent conversation model

In this section, we show when there is a algorithm to recover the hidden graph. We will also show that it is hard to find the likelihood of the maximum likelihood graph for an input sequence of votes. Before we present those two results, we start with the following observation:

Observation 1. *For constant $p \neq 1/2$, under the independent conversation model, it takes exponentially many votes to distinguish with high probability between the complete graph and the complete graph minus an edge.*

This follows directly from the fact that in both the complete graph and the complete graph minus an edge, if $p \neq 1/2$, with exponentially high probability every voter will vote 1. Our only hope is that it becomes possible to recover the graph G when $p = 1/2$, which we show to be the case.

An algorithm for $p = 1/2$

In this section, we prove the following:

Theorem 2. *Let $p = q = 1/2$. For any graph G on n vertices and $\delta > 0$, if $m = \Omega\left(n^2\left(\ln\frac{1}{\delta} + \ln n\right)\right)$ votes are drawn from G under the independent conversation model, there is a polynomial-time algorithm that will recover G with probability at least $1 - \delta$.*

Let $X_u \in \{1, -1\}$ be the random variable representing the outputted vote of vertex u, so $X_u = 1$ if u votes 1 and -1 otherwise. Now consider two vertices u and v. The votes of u and v are independent if and only if (u, v) is not an edge. This yields a natural approach to determining if (u, v) is an edge of G: measure the sample covariance between the votes of u and v and if this covariance is sufficiently far away from zero, there must be an edge.

To formalize this, we need to calculate the covariance between X_u and X_v if there is an edge between them:

Lemma 3. *For any edge (u, v) of G, let d_u and d_v be the degrees of u and v. For convenience, let $\rho = (1 - 2p)q + p$. Then Cov (X_u, X_v) is*

$$\begin{cases} 4\rho^2\binom{d_u-1}{\frac{d_u-2}{2}}\binom{d_v-1}{\frac{d_v-2}{2}}\left(p(1-p)\right)^{\frac{d_u+d_v-2}{2}}, & \textit{even } d_u, d_v \\ 4\rho\binom{d_u-1}{\frac{d_u-2}{2}}\binom{d_v-1}{\frac{d_v-1}{2}}\left(p(1-p)\right)^{\frac{d_u+d_v-1}{2}}, & \textit{even } d_u\textit{, odd } d_v \\ 4\rho\binom{d_u-1}{\frac{d_u-1}{2}}\binom{d_v-1}{\frac{d_v-2}{2}}\left(p(1-p)\right)^{\frac{d_u+d_v-1}{2}}, & \textit{odd } d_u\textit{, even } d_v \\ 4\binom{d_u-1}{\frac{d_u-1}{2}}\binom{d_v-1}{\frac{d_v-1}{2}}\left(p(1-p)\right)^{\frac{d_u+d_v}{2}}, & \textit{odd } d_u, d_v. \end{cases}$$

Proof. Consider an edge $(u, v) \in E(G)$. Since u and v vote independently given the vote of the edge (u, v), we will write the probability that each of these vertices vote 1 given the edge vote.

Namely, call $P^1_u = \mathbb{P}\left(u \text{ votes } 1 | \text{edge } (u, v) \text{ votes } 1\right)$ and $P^{-1}_u = \mathbb{P}\left(u \text{ votes } 1 | \text{edge } (u, v) \text{ votes -}1\right)$ and similarly P^1_v, P^{-1}_v the analogous probabilities for v.

We can write the covariance in terms of these four probabilities: *Cov* $(X_u, X_v) = 4p(1-p)(P^1_u - P^{-1}_u)(P^1_v - P^{-1}_v)$.

To show this, it suffices to write the covariance as a function of the joint probabilities $P(X_u = 1, X_v = 1)$, etc., and then write each joint probability as a function of the probabilities that a vertex votes 1 given that how the adjacent edge votes. For example, by conditioning on the vote of edge (u, v),

$$P(X_u = X_v = 1) = pP^1_uP^1_v + (1-p)P^{-1}_uP^{-1}_v.$$

The others are similar.

To complete the proof, all we need is a formula for P^1_u (P^{-1}_u, P^1_v, and P^{-1}_v are calculated analogously). This is done by choosing edges to form the majority vote of u's neighborhood.

Recall $d(u) - 1$ and $d(v) - 1$ are the degrees of u and v, respectively, minus 1 (in order to discount the edge (u, v)). We then have

$$P^1_u = \begin{cases} \sum_{i=0}^{\frac{d(u)-2}{2}}\binom{d(u)-1}{i}(1-p)^ip^{d(u)-1-i} & \text{even } d(u) \\ +\, q\binom{d(u)-1}{\frac{d(u)}{2}}(1-p)^{\frac{d(u)}{2}}p^{\frac{d(u)-2}{2}}, & \\ \sum_{i=0}^{\frac{d(u)-1}{2}}\binom{d(u)-1}{i}(1-p)^ip^{d(u)-1-i}, & \text{odd } d(u). \end{cases}$$

The statement of the lemma then follows. □

In the case where $p = 1/2$, we show that the covariance will be sufficiently large; namely that it will be $\Omega(1/n)$, where n is the number of vertices of G.

Corollary 4. *When $p = 1/2$ and (u, v) is an edge of G,*

$$Cov\,(X_u, X_v) \geq \frac{1}{2\pi}\frac{1}{\sqrt{d_ud_v}} \geq \frac{1}{2\pi n}.$$

Proof. We simplify the formula for the covariance derived in Lemma 3 by giving lower bounds for the central binomial coefficients, from which the result immediately follows: For any positive integer k, the central binomial coefficient(s) satisfy $\binom{k}{\lceil\frac{k-1}{2}\rceil} \geq \frac{2^k}{\sqrt{\pi k}}$. These lower bounds follow from Sterling's approximation $k! = \sqrt{2\pi k}\left(\frac{k}{e}\right)^k\left(1 + O\left(\frac{1}{k}\right)\right)$. □

Note this lower bound was only polynomial in $1/n$ because $p = 1/2$; otherwise, the exponential term $(p(1-p))^n$, for p constant, ensures that the covariance goes to 0 exponentially quickly in n.

We are now ready to prove Theorem 2, which uses the Hoeffding bound to establish that the sample covariance converges quickly enough to its expectation, which if there is an edge is given in Lemma 3 and if there is no edge is just 0.

proof of Theorem 2. Recall that in the independent conversation model, we are given m votes $X_{u,i} \overset{i.i.d.}{\sim} X_u$ for $i = 1, \ldots, m$ and all u in G, where X_u is the $\{-1, 1\}$-valued random variable found by taking the majority vote of the initial votes of u's neighborhood.

For $p = q = 1/2$, $\mathbb{E}\left(X_u\right) = 0$ for each vertex u, which means that $Cov\left(X_v, X_u\right) = \mathbb{E}\left(X_u X_v\right)$. This means that the (biased) sample covariance between u and v is $C^m_{u,v} = \frac{1}{m}\sum_i^m X_{u,i}X_{v,i}$.

The algorithm to recover G from the m votes is straightforward: For each pair of vertices u, v, calculate the sample covariance $C^m_{u,v}$. If $C^m_{u,v} > \frac{1}{4\pi n}$, then the algorithm claims there is an edge between u and v, and otherwise, the algorithm claims there is no such edge. We call this the *covariance test*. It suffices to show that the probability that the covariance test is wrong is low: Using Corollary 4, we get for edge (u, v),

$$\mathbb{P}\left(C^m_{u,v} < \frac{1}{4\pi n}\right) \leq \mathbb{P}\left(C^m_{u,v} - \mathbb{E}\left(C^m_{u,v}\right) < -\frac{1}{4\pi n}\right),$$

and for $(u, v) \notin E(G)$,

$$\mathbb{P}\left(C^m_{u,v} > \frac{1}{4\pi n}\right) = \mathbb{P}\left(C^m_{u,v} - \mathbb{E}\left(C^m_{u,v}\right) > \frac{1}{4\pi n}\right).$$

From the Hoeffding bound each of these two terms is bounded above by $e^{-\frac{cm}{n^2\pi^2}}$ for some constant c.

Let G' be the network inferred by the above algorithm. Then the probability that G' is not G is no more than

$$\sum_{u,v \in E} \mathbb{P}\left(C^m_{u,v} < \frac{1}{4\pi n}\right) + \sum_{u,v \notin E} \mathbb{P}\left(C^m_{u,v} > \frac{1}{4\pi n}\right),$$

which is bounded from above by $\binom{n}{2}e^{-\frac{cm}{n^2\pi^2}}$.

Hence, for any $\delta > 0$, setting $m = \Omega\left(n^2\left(\ln\frac{1}{\delta} + \ln n\right)\right)$ suffices so that $\binom{n}{2}e^{-\frac{cm}{n^2\pi^2}} < \delta$. □

Hardness of computing the MLE

As we have seen, when $p = q = 1/2$, distinguishing between graphs can be done in polynomial time. This might give hope that, in this case, computing the likelihood of the MLE graph, given a set of votes, may be easy. Alas, we give hardness results indicating this is not the case.

We reduce from Conitzer's problem of computing $\mathbb{P}\left(V^*|G\right)$, where V^* is a vote produced by a given graph G.[1] He shows that this is problem is #P-hard by reducing from counting the number of perfect matchings in a bipartite graph. Surprisingly, our proof of this hardness result uses the easiness of distinguishing graphs in polynomial time in the case when $p = q = 1/2$. This is a key component in Lemma 6. Formally, we prove the following theorem.

Theorem 5. *There is a randomized polynomial-time oracle reduction from computing the MLE of the maximum likelihood graph from a sequence of votes w.h.p. to counting the number of perfect matchings in a balanced bipartite graph.*

First, we need some notation: For any graph G on n vertices, let $\mathcal{V}_G$ be the distribution over a set of n votes induced by G under the independent conversation model for $p = 1/2$. For convenience we will denote the m-product distribution $\mathcal{V}_G \times \ldots \times \mathcal{V}_G$ as $\mathcal{V}_G^{[m]}$. That is, for any vote $V \in \{-1, 1\}^n$, $\mathbb{P}_{\mathcal{V}_G}(V) = \mathbb{P}\left(V|G\right)$ is the probability mass of V under $\mathcal{V}_G$. Similarly, for a sequence of votes $V^{[m]}$, $\mathbb{P}_{\mathcal{V}_G^{[m]}}(V^{[m]}) = \mathbb{P}\left(V^{[m]}|G\right)$ is the probability mass of $V^{[m]}$ under $\mathcal{V}_G^{[m]}$.

Once a large enough sample is drawn from G, we show that not only with high probability G is the MLE for the votes, but also its probability of producing the sample can be made arbitrarily better than that of the second most likely graph. Furthermore, the size of the sample needed in order to have this happen is only polynomially large in the number of vertices.

Lemma 6. *For $\delta > 0$, $\alpha > 1$,*

$$\mathbb{P}_{V^{[m]} \sim \mathcal{V}_G^{[m]}}\left(\mathbb{P}\left(V^{[m]}|G\right) \leq \alpha \max_{G' \neq G} \mathbb{P}\left(V^{[m]}|G'\right)\right) < \delta$$

for $m = \Omega\left(n^2\left(n^2 + \ln\frac{1}{\delta} + \ln\alpha\right)\right)$.

Proof. Fix $\alpha > 1$ and denote by E the event that

$$\frac{\max_{G' \neq G}\mathbb{P}(V^{[m]}|G')}{\mathbb{P}(V^{[m]}|G)} \geq \frac{1}{\alpha}.$$

(If $\mathbb{P}(V^{[m]}|G) = 0$, then the desired event occurs, so we can safely assume the converse.) The idea of this proof is that we will show the probability of E happening is small by conditioning on what the vote sequence $V^{[m]}$ looks like when drawn from G. Specifically, in the proof of Theorem 2, we show that the covariance test would have successfully found G with high probability, so we condition on this happening.

Fix a graph $G' \neq G$. We will want to show that the probability that $V^{[m]}$ is pulled from G' (instead of G) is sufficiently small. The covariance test failed if $V^{[m]}$ were pulled from G': the covariance test returned G on $V^{[m]}$ instead of G'. And again, the probability that the covariance test failed is low, as showed in the proof of Theorem 2, so the probability that $V^{[m]}$ is pulled from G' must be small.

We denote the set of vote sequences for which the covariance test returns G by Φ_G.

Using this notation, we condition on $V^{[m]}$ being in Φ_G or not and then get an immediate upper bound:

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(E\right) \leq \mathbb{P}_{\mathcal{V}_G^{[m]}}\left(E|V^{[m]} \in \Phi_G\right) + \mathbb{P}_{\mathcal{V}_G^{[m]}}\left(V^{[m]} \notin \Phi_G\right).$$

[1] While the problem Conitzer considers is slightly different than computing $\mathbb{P}\left(V^*|G\right)$, in the case where $p = 1/2$, his problem reduces to computing $\mathbb{P}\left(V^*|G\right)$.

We then bound each of these two terms. The probability that the covariance test failed on $V^{[m]}$ is small: By inspecting the proof of Theorem 2, we have that for some constant c,

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(V^{[m]} \notin \Phi_G\right) \leq \binom{n}{2} e^{-\frac{cm}{n^2\pi^2}}. \tag{1}$$

Otherwise, the covariance test succeeded and we condition on $V^{[m]} \in \Phi_G$. We now show that

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(E | V^{[m]} \in \Phi_G\right) \leq \alpha\left(2^{\binom{n}{2}} - 1\right)\binom{n}{2} e^{-\frac{cm}{n^2\pi^2}}. \tag{2}$$

Markov's inequality gives

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(\frac{\max_{G'\neq G}\mathbb{P}\left(V^{[m]}|G'\right)}{\mathbb{P}\left(V^{[m]}|G\right)} \geq \frac{1}{\alpha} \middle| V^{[m]} \in \Phi_G\right) \leq \alpha \cdot \mathbb{E}_{\mathcal{V}_G^{[m]}}\left(\frac{\max_{G'\neq G}\mathbb{P}\left(V^{[m]}|G'\right)}{\mathbb{P}\left(V^{[m]}|G\right)} \middle| V^{[m]} \in \Phi_G\right).$$

It is then enough to expand this expected value using the definition to get that

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(E | V^{[m]} \in \Phi_G\right) \leq \alpha \sum_{V^{[m]} \in \Phi_G} \max_{G' \neq G} \mathbb{P}\left(V^{[m]}|G'\right).$$

Now we group the terms of the sum by which graph $G' \neq G$ maximizes the probability $\mathbb{P}\left(V^{[m]}|G'\right)$. There may be many terms in the sum that any one graph G' maximizes, but certainly each vote sequence associated with each term is in Φ_G. There are of course $2^{\binom{n}{2}} - 1$ such graphs, so

$$\sum_{V^{[m]} \in \Phi_G} \max_{G' \neq G} \mathbb{P}\left(V^{[m]}|G'\right) \leq \sum_{G' : G' \neq G} \sum_{V^{[m]} \in \Phi_G} \mathbb{P}\left(V^{[m]}|G'\right) = \left(2^{\binom{n}{2}} - 1\right)\mathbb{P}\left(V^{[m]} \in \Phi_G | G'\right).$$

If $V^{[m]}$ were in Φ_G but $V^{[m]}$ was pulled from G', then the covariance test has failed at returning G'. So

$$\mathbb{P}\left(V^{[m]} \in \Phi_G | G'\right) \leq \binom{n}{2} e^{-\frac{cm}{n^2\pi^2}},$$

implying Equation 2. Combining Equations 1 and 2, we get

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}(E) \leq \alpha 2^{\binom{n}{2}} \binom{n}{2} e^{-\frac{cm}{n^2\pi^2}}.$$

Thus it suffices to set $m = \Omega\left(n^2\left(n^2 + \ln\frac{1}{\delta} + \ln\alpha\right)\right)$, for $\mathbb{P}_{\mathcal{V}_G^{[m]}}(E)$ to be upper-bounded by δ. □

We now give a sketch of the proof of Theorem 5.

proof sketch of Theorem 5. For convenience, we restrict to the case where all vertices of the graph have odd degree. It suffices to consider the case where $p = 1/2$.

The idea of the proof is going to be to build a sequence of votes $V^{[m]}$ whose MLE we know to be the input G, and then compute $\mathbb{P}(V^*|G) = \frac{\mathbb{P}(V^{[m]}, V^*|G)}{\mathbb{P}(V^{[m]}|G)}$. Our oracle will give us the values of the right-hand side. This approach will work if $\mathbb{P}(V^*|G) \neq 0$.

So we first test for the case if $\mathbb{P}(V^*|G) = 0$. Conitzer provides a way to do this for a similar problem: his reduction is from the maximum weighted b-matching problem, which we can adapt to the so-called "c-capacitated" version that we need [4, 12].

Else, $P(V^*|G) \neq 0$. We draw a sequence of votes $V^{[m]} \overset{i.i.d.}{\sim} \mathcal{V}_G \times \ldots \times \mathcal{V}_G$. Lemma 6 immediately implies that G will be the MLE for $V^{[m]}$ with high probability certainly when $m = \Omega(n^4 \ln n)$. In other words, with just $\Omega(n^4 \ln n)$ votes we will successfully query the oracle for $\mathbb{P}(V^{[m]}|G)$ with high probability.

It suffices to ensure that with high probability we will also successfully query the oracle for the $m+1$-length sequence $V^{[m]}, V^*$. Recall that since $\mathbb{P}(V^*|G)$ is the sum, over all satisfying edge votes, of the quantity $\left(\frac{1}{2}\right)^{|E(G)|}$, where $E(G)$ is the edge set of G and $p = 1/2$. There must be at least one satisfying edge-vote assignment since $\mathbb{P}(V^*|G) \neq 0$, so $\mathbb{P}(V^*|G) \geq \left(\frac{1}{2}\right)^{|E(G)|}$. In addition, again by Lemma 6, for any $G' \neq G, \frac{\mathbb{P}(V^{[m]}|G)}{\mathbb{P}(V^{[m]}|G')} > \alpha$ w.h.p. Then for any $G' \neq G$,

$$\mathbb{P}(V^{[m]}, V^*|G') < \left(\frac{1}{\alpha}\mathbb{P}(V^{[m]}|G)\right)\left(2^{|E(G)|}\mathbb{P}(V^*|G)\right) = \frac{2^{|E(G)|}}{\alpha}\mathbb{P}(V^{[m]}, V^*|G).$$

Finally, $\alpha > 2^{|E(G)|}$ with probability at least $1 - \delta$ when $\delta = o_n(1)$ and $\alpha = \Omega(e^{n^2})$, which in turn implies we need $m = \Omega(n^4 \ln n)$. Since m is polynomial in n, the oracle reduction, once it tests for the existence of at least one valid edge vote, simply consists of drawing a polynomial number of votes from G and then querying the oracle for $\mathbb{P}(V^{[m]}, V^*|G)$ and $\mathbb{P}(V^{[m]}|G)$. □

The common neighbor model

We now turn our attention to the common neighbor model. Again, we ask if it is possible to recover G by seeing only polynomially many votes. In general, it is not possible to recover G at all, let alone with only polynomially many votes:

Observation 7. *Under the common neighbor model, there is no algorithm that can distinguish between matchings.*

If G is a matching between the vertices, each vertex will vote how its neighbor votes, meaning that each vertex votes i.i.d. with probability p regardless. Thus there is no way to distinguish between different matchings.

Recovering A^2 from covariances

Given the impossibility of recovering the graph, we relax the problem to the following: Find a graph that is likely to produce the given votes in the sense that the expected covariances of this graph should be as close as possible (under some norm) to the covariances of the observed votes. This problem is motivated by the algorithm for the independent conversation model which finds a graph whose expected covariances match the measured covariances.

Yet even if we were to know the *expected* covariances of the input votes, finding a graph whose expected covariances are close to those input covariances remains challenging:

Observation 8. *For the common neighbor model, finding a graph with given expected vote covariances is at least as hard as recovering an adjacency matrix from its square.*

To prove this observation, it suffices to show that the expected covariances are a function solely of the entries of A^2. Then recovering A from A^2 consists of using the entries of A^2 to compute the expected covariances, at which point the adjacency matrix of a graph with those covariances will be exactly A. The i, jth entry of A^2 is the number of length-two paths between i and j, so it is enough to write the covariances of a graph in terms of the following: For $\Gamma(v)$ the neighborhood of a vertex v, denote $d_{uv} = |\Gamma(v) \cap \Gamma(u)|$, $d_u = |\Gamma(u) \setminus (\Gamma(v) \cap \Gamma(u))|$, and d_v analogously. The covariances are a function of d_u, d_v, and d_{uv}. For the sake of simplicity we will assume that $|\Gamma(u)|$ and $|\Gamma(v)|$ are odd, but it is straightforward to modify the formula given below in the cases when they are not.

Lemma 9. *Assume* $|\Gamma(u)|$ *and* $|\Gamma(v)|$ *are odd. For* $p = 1/2$,

$$Cov(X_u, X_v) = \frac{1}{2^{d_{uv}-2}} \left(\sum_{k=0}^{d_{uv}} \binom{d_{uv}}{k} P_{u,v}(k) P_{v,u}(k) \right) - 1,$$

where, for $\theta = (d_{uv} + d_u + 1)/2 - k$,

$$P_{u,v}(k) = \begin{cases} \frac{1}{2^{d_u}} \sum_{i=\theta}^{d_u} \binom{d_u}{i} & \text{if } 0 \le \theta \le d_u \\ 0 & \text{if } \theta > d_u \\ 1 & \text{if } \theta \le 0. \end{cases}$$

Proof. Let X_u represent vertex u's vote. When $p = 1/2$, $E[X_u] = 0$, and $P(X_u = X_v = 1) = P(X_u = X_v = 0)$, so the covariance $Cov(X_u, X_v)$ is

$$E[X_u X_v] = 2P(X_u = X_v) - 1 = 4P(X_u = X_v = 1) - 1.$$

To determine $P(X_u = X_v = 1)$, we condition on the number of common neighbors that voted 1:

Assuming some k common neighbors vote 1, in order for u to vote 1, u needs an additional $\frac{d_{uv}+d_u+1}{2} - k$ neighbors to vote 1. If k is already at least $\frac{d_{uv}+d_u+1}{2}$, then the probability of voting 1 is already 1; on the other hand if there aren't enough remaining vertices to vote 1, then the probability is 0. This yields $P_{u,v}(k)$ as the probability that u votes 1 given that k common neighbors of u and v voted 1.

Now we can write $P(X_u = X_v = 1)$ as

$$\sum_{k=0}^{d_{uv}} \binom{d_{uv}}{k} p^k (1-p)^{d_{uv}-k} P_{u,v}(k) P_{v,u}(k),$$

completing the proof. □

When recovering a graph from a sequence of input votes, we are not even given the expected covariances of the input votes. Instead we can calculate the measured covariances, from which we can determine A^2. At this point, we have a function inversion problem on our hands: We can find A^2 merely by recovering these d_{uv}'s and d_u's from the covariances, but given that the formula given in Lemma 9 is not closed, this is not trivial. Since there are only polynomially many possible values for d_{uv} and d_u, we can simply try all values to find the covariance closest to the observed value. However, there may be covariances that are exponentially close to each other, making it impossible to distinguish between these values for given d_{uv}, d_u. In this case the values recovered for the entries of A^2 may not be unique, which in the worst case leads to the generalized squared adjacency problem. Even in the case when we recover unique values, it still reduces to the squared adjacency problem.

While the squared adjacency problem is open, we show:

Theorem 10. *The generalized squared adjacency problem is* NP*-hard.*

Proof. The reduction is from CLIQUE, which asks if there is a clique of size k on the input graph. Given a graph $G = (V, E)$ and an integer k, we construct a set system $\{S_{ij}\}$ with $(n+1)^2$ sets, where $n = |V|$. We then show that G has a clique of size k if and only if there is a graph $G' = (V \cup \{v\}, E')$ such that the i, jth entry of $A(G')^2$ is in $S_{i,j}$, where $A(G')$ is the adjacency matrix of G'. The $(n+1)^2$-sized set system $\{S_{i,j}\}$ is defined as follows:

$$S_{ij} = \begin{cases} \{0,1\} & \text{if } i \neq j \text{ and } i,j \neq v \text{ and } (i,j) \in E(G) \\ \{0\} & \text{if } i \neq j \text{ and } i,j \neq v \text{ and } (i,j) \notin E(G) \\ \{0\} & \text{if } i = v \text{ and } j \neq v \\ \{0\} & \text{if } j = v \text{ and } i \neq v \\ \{k\} & \text{if } i = j = v \\ \{0,1\} & \text{if } i = j \text{ and } i,j \neq v \end{cases}$$

Assume there is a clique of size k in G. Then G' is defined as follows: Denote the vertex set of the clique in G by C. G' will have an edge between v and all members of C, and no other edges. It is straightforward to check that the i, jth entry of $A(G')^2$ is in S_{ij} by noting that the diagonal entries of $A(G')^2$ are the vertices' degrees and the off-diagonal entries counts the number of common neighbors.

In the other direction, assume there is such a graph G' whose squared adjacency matrix satisfies the constraints imposed by the set system $\{S_{i,j}\}$. In this case, the clique of size k in G will be exactly the neighborhood of v in G' (not including v itself). Call $N(v)$ the neighborhood of v in G'. Note the degree of v in G' must be k, by definition of S_{vv}, i.e. $|N(v)| = k$. Consider a distinct pair of vertices i, j in $N(v)$. The vertices i and j have at least one common neighbor in G', namely v, because both are in $N(v)$, meaning that $A(G')^2_{ij} \geq 1$. But if (i, j) is not an edge in G then $S_{ij} = \{0\}$ by the definition of S_{ij}, a contradiction, forcing $N(v)$ to be a clique as required. □

A heuristic approach

Unlike in the independent conversation model, we have no efficient algorithm for producing the social network under the common neighbor model. Hence, we employ a heuristic to find a graph that satisfies or comes close to satisfying the constraints imposed on it by the measured covariances.

Because of the computational hardness of this problem, we propose a heuristic approach to learn networks under the common neighbor model. This heuristic will be used in our experimental results in Section . Our heuristic, Algorithm 1, finds those pairs of vertices whose current expected covariance (assuming $p = 1/2$) is farthest away from the measured covariance and modifies the graph to decrease that gap.

Algorithm 1 Common neighbor heuristic

```
Input: $\{\hat{c}_{ij}\}$, measured covariances between voters $i$ and $j$.
Input: $T$, number of iterations to run.
  $G \sim \mathcal{G}(n, 1/2)$, $G_{best} := \emptyset$
  $\{c_{ij}\} := \{\sigma(i,j)\}$    # calculate expected covariances
  for $0$ to $T$ do
    $i, j :=$ argmax$_{i,j}|c_{ij} - \hat{c}_{i,j}|$
    $G :=$ Modify$G(G, i, j, \hat{c}_{ij}, c_{ij})$
    $\{c_{ij}\} := \{\sigma(i,j)\}$    # update the expected covariances
    if $\sum_{i,j} c_{ij} - \hat{c}_{ij}$ is smallest so far then $G_{best} := G$
  end for
  return $G_{best}$
```

Algorithm 2 Modify$G$

```
Input: Graph $G$; vertices $i, j$; $\hat{c}_{ij}$, $c_{ij}$ the measured and expected covariances between $i$ and $j$.
  $unconnected := V(G) \setminus (\Gamma(i) \cup \Gamma(j) \cup \{i, j\})$
  $cn := \Gamma(i) \cap \Gamma(j)$
  if $c_{ij} - \hat{c}_{ij} > 0$ then
    randomize among whichever of these are available:
    1: x := random($i, j$), y := random(unconnected)
        add edge $(x, y)$ to $G$
    2: x := random($i, j$), y := random(cn).
       delete edge $(x, y)$ from $G$
    3: y := random(cn)
       delete edges $(i, y)$ and $(j, y)$ from $G$
  else if $c_{ij} - \hat{c}_{ij} < 0$ then
    randomize among whichever of these are available:
    1: y := random(unconnected)
       add edges $(i, y)$ and $(j, y)$ to $G$
    2: y := random($\Gamma(i) \setminus cn$)
       add $(i, y)$ or delete $(j, y)$ from $G$ randomly
    3: y := random($\Gamma(j) \setminus cn$)
       add $(j, y)$ or delete $(i, y)$ from $G$ randomly
  end if
  return $G$
  # where random(.) selects an element of its input u.a.r.
```

The local changes we want to make clearly cannot just consist of adding/removing single edges: Say the covariance between a given pair of vertices needs to go up and those vertices' neighborhoods are currently empty. The only way to increase the covariance is to add at least two edges: (i, v) and (v, j) for some other vertex v. The natural compromise is then to add or remove the minimal number of edges (either one or two) to change the covariance, as seen in Algorithm 2.

Experimental Results

In this section, we test our algorithms on United States Senate roll call votes. We examine each two-year congressional session as one voting network. Each Senate member is an agent who either votes for the bill in question, against, or does not vote (either because the senator served only part of the term or because the senator just didn't vote on the bill), yielding votes from the set $\{-1, 0, 1\}$.

Obviously, our models are simplifications — they don't take into account evolution of opinion, nor do they take into account the possibility of anti-correlated voters. Even assuming that either model is representative, when presented real data, the parameter p is not given, as is assumed above. The algorithms we present assume that $p = 1/2$, which is not necessarily the case. Finally, we are given a fixed amount of data, independent of the number of voters.

Despite these limitations, for the independent conversation model, our covariance test, which forms the basis for Theorem 2, results in intuitive behavior. Note that while our model assumes binary votes, our covariance test is general enough to handle such votes — covariance is calculated between the $\{-1, 0, 1\}$-valued votes and the threshold remains the same as in the original covariance test.[2] Examples of the results from this covariance test on the US Senate are shown in Figure 3. Given the highly structured nature of these graphs, it is possible to recover senators' places on the left/right political spectrum, but since this is not the focus of this paper, we do not go into any further detail here.

For the common neighbor model, we use Algorithm 1. Examples of results of this heuristic run on US Senate data are shown in Figure 4. Graphs under this model appear to be very different from those found using the covariance test under the independent conversation model.

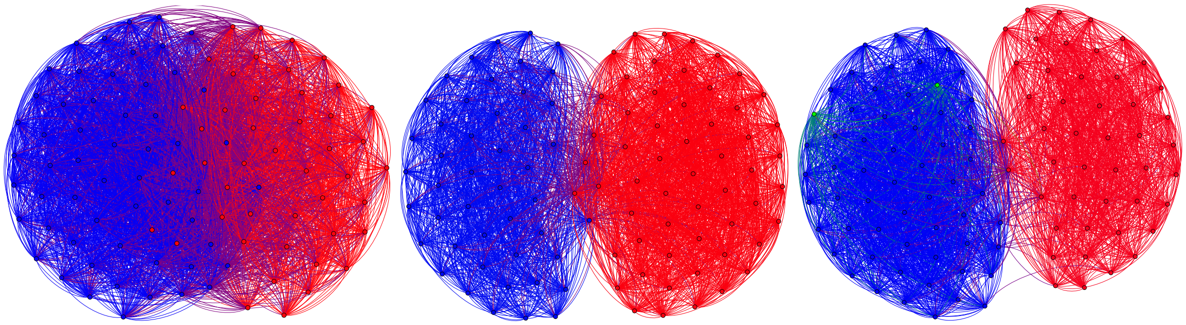
To demonstrate these marked differences, in Figure 5 we give modularity values and average degrees of Democrats and Republicans under both models for the period 1989-2014 (corresponding to the 101st through 113th Congresses). Modularity is a standard measure of the amount of division between communities [11]. Both average degree and modularity are much higher under the independent conversation model (dashed lines in Figure 5) than under the common neighbor model (solid lines). Since the heuristic is randomized, we average these statistics over twenty graphs, each of which is an independent run of the heuristic with 100,000 rounds.

Conclusion

In this paper we derive algorithms and lower bounds for recovering graphs from their vertices' votes under two distinct models. We also present experiments on the U.S. Senate voting network. In the independent conversation model, we show when the graph is recoverable using only a polynomial number of votes. However, if we want to instead take a maximum likelihood approach to recovering graphs, then the task becomes computationally hard.

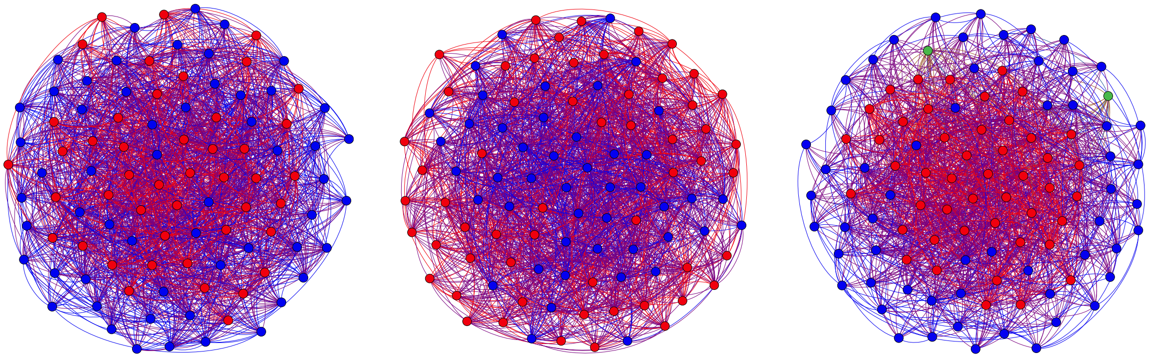
The common neighbor model, on the other hand, leads to significantly different results. Not only is it impossible to recover the graph using only polynomially many votes, finding a graph whose votes' covariances are close to the observed

[2] We do, however, use the unbiased sample covariance instead of the biased sample covariance, as the assumption that $p = 1/2$ no longer necessarily holds, despite the analysis of the algorithm assuming it.



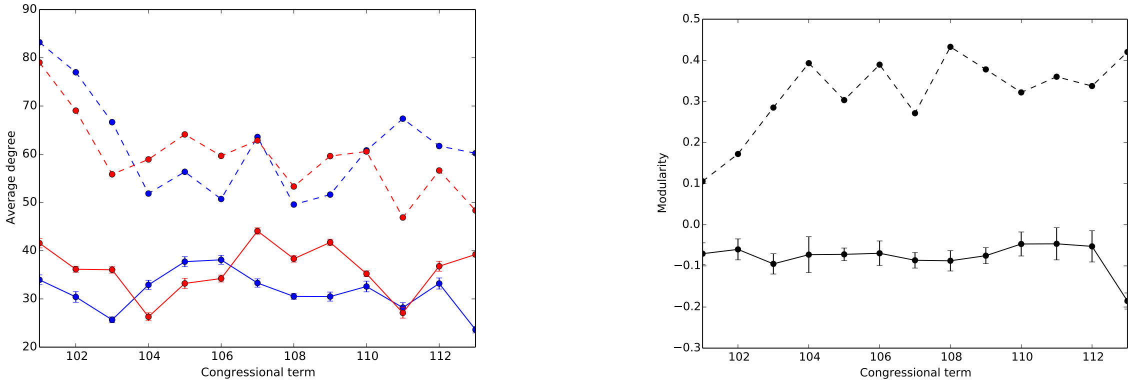
(a) 101st Congress (1989-1990) (b) 106th Congress (1999-2000) (c) 113th Congress (2013-2014)

Figure 3: Graphs of the US Senate for three congressional terms under the independent conversation model. Democrats are colored blue, Republicans are red, and Independents are green.



(a) 101st Congress (1989-1990) (b) 106th Congress (1999-2000) (c) 113th Congress (2013-2014)

Figure 4: Graphs of the US Senate for three congressional terms under the common neighbor model. Democrats are colored blue, Republicans are in red, and Independents are in green.



(a) Average degree. The blue and red lines are the average degree of the Democrats and the Republicans, respectively.

(b) Modularity between Democrats and Republicans. (Independents are included with the party with which they caucus.)

Figure 5: Data for the 101st-113th Congress. Dashed and solid lines are statistics for the independent conversation model and common neighbor model, respectively. Error bars represent one standard deviation, over 20 trials.

covariances leads to having to solve a hard problem (the generalized squared adjacency problem).

This implies that these models really are very different from each other, despite their very similar definitions. This is strong evidence that much care needs to be taken when choosing voting models for network inference. Experiments on Senate roll call data support this conclusion.

Acknowledgements

This research was funded in part by Army Research Office grant #66497-NS under the Young Investigator Program. We thank Vincent Conitzer, Jeremy Kun, Madhu Sudan, and Avi Wigderson for helpful discussions.

References

[1] A. Amelio and C. Pizzuti. Analyzing voting behavior in Italian Parliament: Group cohesion and evolution. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2012*, pages 140–146. IEEE, 2012.

[2] C. Andris, D. Lee, M. J. Hamilton, M. Martino, C. E. Gunning, and J. A. Selden. The rise of partisanship and super-cooperators in the US House of Representatives. *PLoS ONE*, 10(4), 2015.

[3] W. Chen, L. V. Lakshmanan, and C. Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[4] V. Conitzer. The maximum likelihood approach to voting on social networks. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pages 1482–1487. IEEE, 2013.

[5] S. Frischknecht, B. Keller, and R. Wattenhofer. Convergence in (social) influence networks. In *Distributed Computing*, pages 433–446. Springer, 2013.

[6] E. Goles and J. Olivos. Periodic behaviour of generalized threshold functions. *Discrete Mathematics*, 30(2):187–189, 1980.

[7] A. Jakulin, W. Buntine, T. M. La Pira, and H. Brasher. Analyzing the US Senate in 2003: Similarities, clusters, and blocs. *Political Analysis*, 17(3):291–310, 2009.

[8] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015.

[9] K. T. Macon, P. J. Mucha, and M. A. Porter. Community structure in the united nations general assembly. *Physica A: Statistical Mechanics and its Applications*, 391(1):343–361, 2012.

[10] R. Motwani and M. Sudan. Computing roots of graphs is hard. *Discrete Applied Mathematics*, 54(1):81–88, 1994.

[11] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[12] M. Penn and M. Tennenholtz. On multi-object auctions and matching theory: Algorithmic aspects. In *Graph Theory, Combinatorics and Algorithms*, pages 173–188. Springer, 2005.

[13] M. A. Porter, P. J. Mucha, M. E. Newman, and A. J. Friend. Community structure in the United States House of Representatives. *Physica A: Statistical Mechanics and its Applications*, 386(1):414–438, 2007.

[14] A. D. Procaccia, N. Shah, and E. Sodomka. Ranked voting on social networks. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2040–2046, 2015.

[15] K. Schaul. A quick puzzle to tell whether you know what people are thinking, October 2015. [The Washington Post; posted online 09-October-2015].

[16] A. Tsang, J. A. Doucette, and H. Hosseini. Voting with social influence: Using arguments to uncover ground truth. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1841–1842, 2015.

[17] A. S. Waugh, L. Pei, J. H. Fowler, P. J. Mucha, and M. A. Porter. Party polarization in Congress: A network science approach. *arXiv preprint arXiv:0907.3509*, 2009.

[18] Y. Zhang, A. Friend, A. L. Traud, M. A. Porter, J. H. Fowler, and P. J. Mucha. Community structure in Congressional cosponsorship networks. *Physica A: Statistical Mechanics and its Applications*, 387(7):1705–1712, 2008.