Model Visualization Techniques for a Social Network Model

Sam Tyner
Department of Statistics and Statistical Laboratory, Iowa State University
and
Heike Hofmann
Department of Statistics and Statistical Laboratory, Iowa State University

August 15, 2018

Abstract

Stochastic actor-oriented models (SAOMs), first introduced by Snijders (1996), are a type of statistical network model for dynamic social networks. Unlike other network models, SAOMs are not very well understood. We use model visualization techniques introduced in Wickham et al (2015) in order to gain a better understanding of the models and their behavior. Viewing the model in the data space places these models into the field of network drawing, while exploring collections of models helps with parameter interpretation. Finally, delving into the fitting algorithms provides more insight into the underlying mechanisms. With the help of static and dynamic visualizations, we bring the hidden SAOM fitting processes into the foreground, eventually leading to a better understanding and higher accessibility of SAOMs for social network analysts.

Keywords: social network analysis, dynamic networks, network mapping, animation, statistical graphics

Contents

1 Introduction & Background

Social networks, such as collaboration networks between academic researchers or friendship networks have been studied for decades. Statistical models such as exponential random graph models (ERGM) and latent space models (LSM) are often used when studying one observation of a network when trying to model its structure (Frank and Strauss, 1986; Hoff et al., 2002). Social networks can also be observed at many points in time, and ERGM or LSM may not be appropriate for modelling the changes because they do not model network change as time passes. When studying *dynamic networks* observed at many time points, we need a model that allows for variation over time. Models for dynamic social networks hold a great deal of potential because of how realistic they can be: social networks evolve over time as edges are formed and dissolved and new nodes join. Modeling the underlying mechanisms that create network changes in time is very complex but also provides potential to uncover hidden truths.

One set of models for dynamic social networks, are stochastic actor-oriented models (SAOMs), introduced by Snijders (1996). These models are different from other dynamic network models because they allow us to add node level information into the model with the traditional network structure information. Adding node-level information into the model is a more intuitive approach to model changes in a social network because, for example we expect people with common interests to be more likely to form relationships. SAOMs allow us to incorporate this information on shared interests in the modeling process.

Unlike other network models, SAOMs are not very well understood. They are relatively new, especially compared to the classic ERGMs, and they are not very tractable analytically. Likelihood functions quickly become very complex due to the dependency structure inherent in the data, and so computational methods, such as Markov chain Monte Carlo (MCMC) are used to fit models. SAOMs are typically fit to dynamic social network data using a series of MCMC phases for finding method of moments (MoM) estimates of model parameters. In order to estimate the parameters of SAOMs, we use the software SIENA, and its `R` implementation `RSiena` by Ripley et al. (2016a). This software is a considerable contribution to the field of social network analysis, but the parameter estimation process is largely hidden from the user. In order to better understand the model-fitting process of RSiena, we visualize the underlying methods and structures, and utimately aim to help researchers working with SAOMs better understand the structure of and results from these models.

To learn more about the SAOM fitting process, we use model visualization techniques (see Wickham et al. (2015)) to display the model in the data space, view collections of models instead of single models, and visually explore the algorithms fitting the SAOMs. SAOMs are a model family that can reap many benefits from the application of model visualization techniques. For instance, the models themselves include a continuous-time Markov chain (CTMC) that is completely hidden from the analyst in the model fitting process. Visualizing the CTMC can provide researchers with more insight into the underlying features of the model. In addition, SAOMs can include a great deal of parameters in the model structure, each of which is attached to a network statistic. These statistics are often somewhat, if not highly, correlated, which causes high correlation between the corresponding parameters in a SAOM. By visualizing collections of SAOMs, we gain a better understanding of these correlations and can find ways to account for their effects in the model. Furthermore, the estimation of the parameters in a SAOM rely on convergence checks based on simulations from the fitted model. These simulations are hidden and unsaved by default for computational efficiency. With the help of both static and dynamic visualizations, we bring the hidden model fitting processes into the foreground, eventually leading to a better understanding and higher accessibility of stochastic actor-oriented models for social network analysts.

1.1 Networks and their Visualizations

We provide a brief introduction to the structure of network data and common network visualization methods.

Network Data Structure: Network data across fields have similar data structures. There are always units of observation and connections between those units. In a social network, the units of observations, called *nodes* or *actors*, may be people, while the connections, called *edges* or *ties*, are the relationships between people. Networks often change over time, like when new relationships are formed in a social network. The nodes and edges themselves can also have inherent variables of interest, e.g. the age, gender, and political affiliation of people in a social network and the type of relationship (friends, married, etc.).

Visualizing Network Data: Network visualization, also called network mapping, is a prominent subfield of network analysis. Visualizing network data is uniquely difficult because of the structure of the data itself. Most data visualizations rely on well-defined axes inherited from the data, such as Cartesian coordinates or spatial locations, but network data typically do not have an inherent structure.

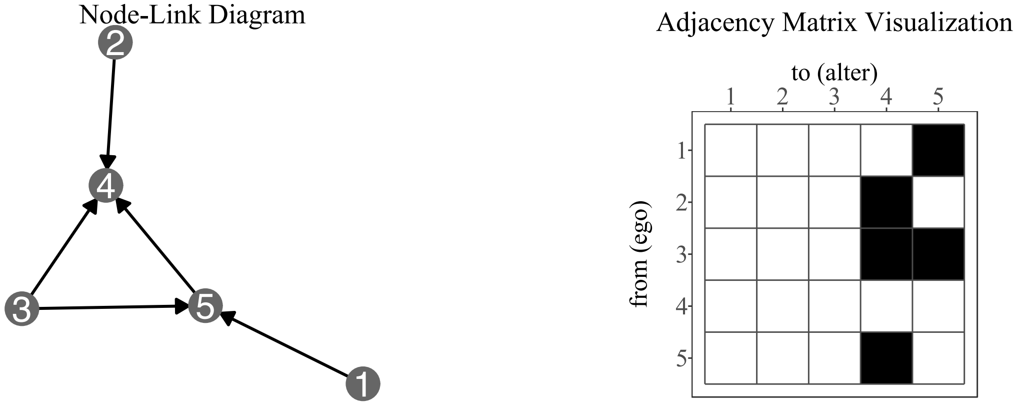


Figure 1: On the left, a node-link diagram of our directed toy network, with nodes placed using the Kamada-Kawai algorithm. On the right, the adjacency matrix visualization for that same network.

There are two primary methods used to visualize networks: the node-link diagram and the adjacency matrix visualization (Knuth, 2013; Fekete, 2009). For example, consider a network with five nodes, $\{1, 2, 3, 4, 5\}$, connected by five directed edges: $\{2 \to 4, 3 \to 4, 1 \to 5, 3 \to 5, 5 \to 4\}$. This data is shown using the two visualization methods in Figure 1.

The first method, the node-link diagram, represents nodes with points in two dimensions and then represents edges by connecting the points with lines. These lines can also have arrows on them indicating the direction of the edge for directed networks, as show in Figure 1. Because there is usually no natural placement of the points, a random placement of the points is used, then adjusted with a layout algorithm, of which there are many (Gibson et al., 2013). Some commonly used algorithms, such as the Kamada-Kawai (KK) layout

(Kamada and Kawai, 1989) and the Fruchterman-Reingold (FR) layout (Fruchterman and Reingold, 1991), are designed to mimic physical systems, drawing the graphs based on the "forces" connecting them. In Figure 1, we use the KK algorithm, and unless otherwise stated, all other node-link diagrams in this paper will use the KK layout.

The second method for network visualization uses the adjacency matrix of the network. The adjaceny matrix of a network, $\mathbf{A}$ is defined as follows: an entry A_{ij} of $\mathbf{A}$, for two nodes $i \neq j$ in the network is defined as

$$A_{ij} = \begin{cases} 1 & \text{if an edge exists } i \rightarrow j \\ 0 & otherwise \end{cases} \tag{1}$$

In this paper, we only consider the presence or absence of an edge, but if the network has weighted edges, A_{ij} is the weight of the edge from $i \rightarrow j$. An adjacency matrix visualization for our toy example is also shown in Figure 1. This visualization shows the network as a grid, with each row and column representing a node. The grid space for edge $i \rightarrow j$ is filled in if $A_{ij} = 1$ and is empty if $A_{ij} = 0$.

Each visualization comes with its own advantages and disadvantages. For instance, paths between two nodes in a network are easier to determine with node-link diagrams than with adjacency matrix visualizations (Ghoniem et al., 2005). In node-link diagrams, node-level information can be incorporated into the visualization by coloring or changing the shape of the points, and edge-level information can be incorporated by coloring the lines, or changing their thickness or linetype. Incorporating a node-level variable into an adjacency matrix visualization is not as straightforward or simple, because this type of visualization features the edges. Adjacency matrix visualization can, however, be useful when the network is very complex, dense, or large (Ghoniem et al., 2005).Ultimately, there is no one "correct" way to visualize network information, and we will be using both the node-link and adjacency matrix visualization methods throughout this paper to explore social networks and stochastic actor-oriented models.

1.2 Defining SAOMs

A Stochastic Actor-Oriented Model (SAOM) is a model that incorporates all three data levels of dynamic networks: actor, tie, and temporal information. It models the change of a network in time, while accounting for changes in network structure due to actor-level covariates. The two titular properties of SAOMs, stochasticity and actor-orientation, are key to understanding networks as they exist naturally. Most social networks, even holding constant the set of actors over time, are ever-changing as relationships decay or grow in seemingly random ways. Actors in social networks also have inherent properties that could affect how they change their role within the network, and existing ties may affect formation or dissolution of other ties.

1.2.1 Definitions, Terminology, and Notation

We use the term *dynamic network* to refer to a network, consisting of a fixed set of n nodes, that is changing over time, and is observed at M discrete time points, $t_1, \dots, t_M$ with $t_1 < t_2 < \cdots < t_M$. We denote the network observation at timepoint t_m by $x(t_m)$ for $m = 1, \dots M$. The SAOM assumes that this longitudinal network of discrete observations is embedded within a CTMC, which we will denote by $X(T)$. This CTMC is almost entirely unobserved: we assume that the beginning of the process, $X(0)$, is equivalent to the first network observation $x(t_1)$, while the end of the process, $X(\infty)$, is equivalent to the last observation $x(t_M)$. Nearly all other steps in the chain are unseen, with the exception of $x(t_2), \dots, x(t_{M-1})$. Unlike the first and last observations of the network, these "in-between" observations do not have direct correspondence with steps in the continuous time Markov chain. Thus, the observations $x(t_2), \dots, x(t_{M-1})$ are considered to be "snapshots" of the network at some point between two steps in the CTMC.

The CTMC process $X(T)$ is a series of single tie changes that happen according to the SAOM's pre-defined rate function, where one actor at a time is selected and it adds or removes one outgoing tie, or does nothing. Once an actor is chosen at random according to the rate function, it is "given" the chance to change a tie, and it tries to maximize its utility function based on the current and currently reachable states of the network.

The Rate Function In general, each actor i in the network x gets an opportunity to changes its ties, x_{ij}, to other nodes $j \neq i$ according to the SAOM's *rate function*, $\rho(x, \mathbf{z}, \boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ are the parameters in the function ρ, and x is the current network state, with covariates of interest $\mathbf{z}$. With this general formulation, it is possible that each node will have a different rate of change. We assume a simple rate function, $\rho(x, \mathbf{z}, \boldsymbol{\alpha}) = \alpha_m$ that is constant across all actors between time t_m and t_{m+1}. With this simple rate function, we have just one set of rate parameters in the overall model. In this simple model, the rate parameters dictate how often an actor i "gets an opportunity" to change one of its ties, x_{ij}, for $j \in \{1, \ldots, n\}$ in the time period from t_m to t_{m+1} for $m = 1, \ldots, M-1$ (Note that if $j = i$, no change is made.) A SAOM also assumes that the actors i are conditionally independent given their ties, $x_{i1}, \ldots, x_{in}$ at the current network state. Let $\tau(i|x, m)$ be the wait time until actor i gets to the opportunity to change its current set of ties. For any time point, T, where $t_m \leq T < t_{m+1}$, the waiting time to the next change made by actor i is exponentially distributed with expected value α_m^{-1}, as shown in Equation 2.

$$\tau(i|x, m)|x_{i1}(m), \ldots, x_{in}(m) \overset{\text{iid}}{\sim} Exp(\alpha_m) \tag{2}$$

From Equation 2, we can derive the waiting time to the next change opportunity by *any* actor in the network, $\tau(x)$. The value $\tau(x)$ is also exponentially distributed with expected value $(n\alpha_m)^{-1}$, where $n\alpha_m$ is the rate at which any tie change occurs. The estimation of this parameter in `RSiena` is simple: the method of moments is used to estimate the rate with the statistic

$$C = \sum_i \sum_j |x_{ij}(t_{m+1}) - x_{ij}(t_m)| \tag{3}$$

which is the total number of changes from observation at time t_m to the observation at time t_{m+1}.

The Objective Function Because of the conditional independence assumptions given in Equation 2, we can consider the objective function for each node separately, as only one tie from one node is changing at a time in the CTMC. The node i, which is given the opportunity to change at the current time point, is called the *ego* node. It has the potential to interact with all other nodes in the network, $j \neq i$. These nodes j, are referred to as *alter* nodes, and do not change any of their ties when i may change. For the ego node i in the current network state x, its *objective function*, which it tries to maximize, is written as

$$f_i(\boldsymbol{\beta}, x, \mathbf{Z}) = \sum_{k=1}^{K} \beta_k s_{ik}(x, \mathbf{Z}), \tag{4}$$

for $x \in \mathcal{X}$, the space of all possible directed networks with the n nodes, and $\mathbf{Z}$, the matrix of covariates, and K the total number of parameters in the objective function. The vector $\boldsymbol{\beta}$ contains the parameters of the model with corresponding statistics, $s_{ik}(x, \mathbf{Z})$, for $k = 1, \ldots, K$. To increase the value of f_i, i will destroy one tie where $x_{ij} = 1$, create one tie where $x_{ij} = 0$, or make no change if all changes will result in lower values of $f_i(\boldsymbol{\beta}, x, \mathbf{Z})$.

According to Snijders (2001, p. 371), there should be at least two parameters included in the model: β_1 for the outdegree of nodes, and β_2 for the reciprocity of nodes. (These effects are also used in ERGMs.) The outdegree represents the propensity of nodes with a lot of outgoing ties to form more outgoing ties (the "rich get richer" effect), and the reciprocity parameter measures the tendency of outgoing ties to be returned within a network. The statistics corresponding to these two parameters, $s_{i1}(x)$ and $s_{i2}(x)$ are given inTable 1. In the `RSiena` software, there are over 80 possible $\boldsymbol{\beta}$ parameters to add to the model. The formulas for the corresponding statistics for all effects are provided in Ripley et al. (2017).

The parameters of $f_i(\boldsymbol{\beta}, x, \mathbf{Z})$ are either structural or covariate parameters. The structural parameters, such as the outdegree and reciprocity parameters, correspond to statistics that are *only* functions of the current network state, x. They model underlying mechanisms of network change, answering questions such as, "How does the existing network structure influence change in the network?" The covariate parameters are functions of x and additional node-level covariates of interest, $\mathbf{Z}$. Some possible covariate parameters and their corresponding statistics are discussed in Section 2 and shown in Figure **??**

The objective function also defines the probability that actor i, when it is the ego node, chooses to change the tie to actor j, denoted by $p_{ij}(\boldsymbol{\beta}, x, \mathbf{Z})$. Let $x(i \rightsquigarrow j)$ be the network identical to x with the exception of tie x_{ij}, which is equal to $1 - x_{ij}$. The probility that the tie x_{ij} changes is defined as:

$$p_{ij}(\boldsymbol{\beta}, x, \mathbf{Z}) = \frac{\exp\{f_i(\boldsymbol{\beta}, x(i \rightsquigarrow j), \mathbf{Z})\}}{\sum_{h \neq i} \exp\{f_i(\boldsymbol{\beta}, x(i \rightsquigarrow h), \mathbf{Z})\}} \tag{5}$$

When the value of the objective function is high at the current state, the probability of not

Structural Effects

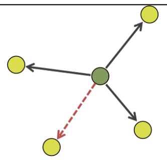
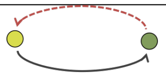
Name	Statistic	Figure
outdegree	$s_{i1}(x) = \sum_j x_{ij}$	
reciprocity	$s_{i2}(x) = \sum_j x_{ij}x_{ji}$	

Table 1: Examples of structural effects included in a SAOM. The dark nodes in the Figures are the ego nodes, and all others are alters. The dotted lines are ties the ego node is en(dis)couraged to make when the parameter corresponding to each effect is positive (negative).

making a change is also high. In the CTMC, when actor i may make a change, it chooses which tie $x_{i1}, \ldots, x_{in}$ to change at random according to the probabilities $p_{ij}(\boldsymbol{\beta}, x, \mathbf{Z})$. The objective function and the resulting values of p_{ij} are combined with the rate function to fully describe the CTMC that is used to model network change in a SAOM. For more on CTMCs, see Yin and Zhang (2010). We explore these probabilities further in Section 5

1.2.2 Fitting Models to Data

To fit a SAOM to dynamic network data, we apply the `RSiena` software, which uses simulation methods to estimate parameter values using either MoM or maximum likelihood (ML) estimation (Ripley et al., 2016a). We chose to use MoM estimation, so we describe that fitting process here, and more information on ML estimation can be found in Snijders (2016).

The underlying SIENA software uses a Robbins-Monro algorithm (see Robbins and Monro (1951)) to estimate the solution of the moment equation

$$E_\theta S = s_{obs} \tag{6}$$

where $\theta = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the vector of rate and objective function parameters, and s_{obs} is the observed vector of the model statistics over all network observations. The full SIENA MoM estimation algorithm operates in three phases, as described in Ripley et al. (2017); Snijders (2016). The first phase performs initial estimation of the score functions for use in the

Robbins-Monro procedure for MoM estimation. The second phase carries out the Robbins-Monro algorithm and obtains estimates of the parameter values through iterative updates and simulation from the CTMC at iterative parameter values. The third phase uses the parameter vector estimated in phase two to estimate the score functions and covariance matrix of the vector of parameter estimates and carries out convergence checks.

In each of the the first two phases of the algorithm, the procedure uses "microsteps" that are simulated from the model as it exists in its current state in order to update either the score functions or the parameter estimates. These simulated microsteps are realized steps in the CTMC that is the backbone of the SAOM. In Section 5, we further explore these phases in the SIENA method-of-moments algorithm through visualization, bringing them out of their "black box" and into the light.

2 Example Data and Models

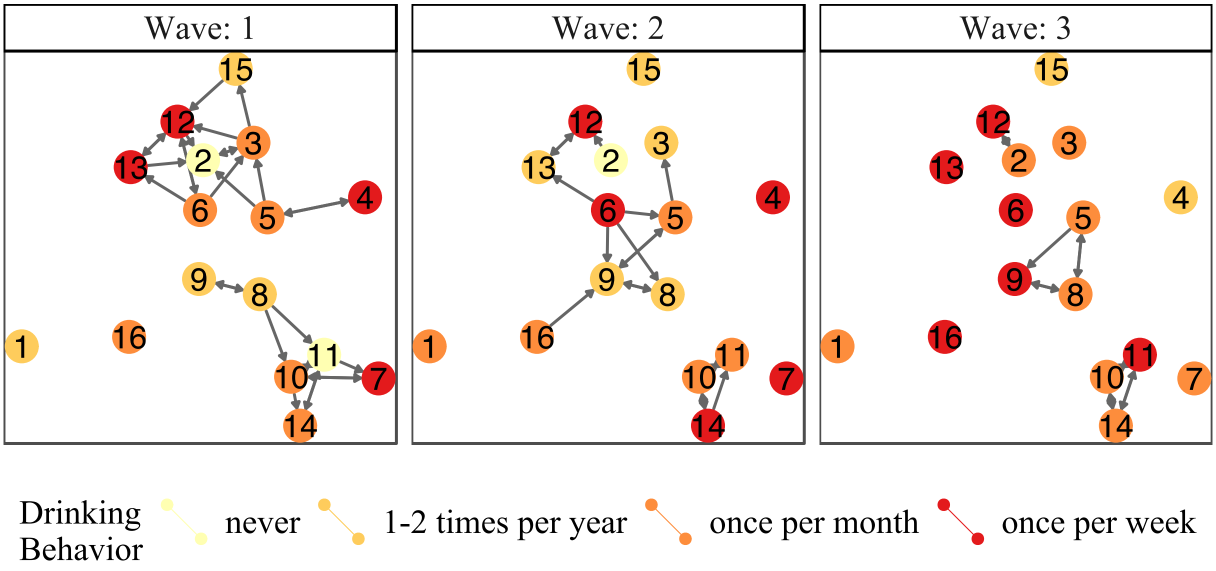


Figure 2: The small friendship network data we will be modelling throughout this paper. The nodes change color as their drinking behavior changes, and the network becomes less densely connected over time. These are the mechanisms we hope to incorporate and model with SAOMs.

Example Data: To guide our visual exploration of SAOMs, we use two example data sources. The first is a subset of the 50 actor dataset from the "Teenage Friends and

Lifestyle Study" that is provided on the RSiena webpage. These data come from Michell and Amos (1997), and we chose to only work with a subset of the data to make node-link diagrams easier to see and to make any changes in the network more noticeable. The node-link visualization of this data is provided in Figure 2. For model fitting, we condition on wave 1 and estimate the parameters of the models for the second and third waves. The actor-level categorical covariate, drinking behavior, has four values: (1) does not drink, (2) drinks once or twice a year, (3) drinks once a month, and (4) drinks once a week. In Figure 2, the nodes are colored according to the drinking behavior of that student. Over time, we can see that the students tend to drink more and become increasingly isolated into smaller groups. An analysis of this type of data with a SAOM should capture these dynamics in a way that allows the researcher to draw conclusions about the nature of the network and behavioral forces at play.

The second data example we use is a collaboration network in the United States Senate during the 111^{th} through 114^{th} Congresses. These sessions of congress correspond to the years of Barack Obama's presidency, from 2009-2016. (Details of how this data can be downloaded are provided by Franois Briatte at `https://github.com/briatte/congress`). In this network, ties are directed from senator i to senator j when senator i cosponsors the bill that senator j authored. There are hundreds of ties between senators when they are connected in this way, so we simplify the network by computing a single value for each pair of senators called the *weighted propensity to cosponsor* (WPC). This value is defined in Gross et al. (2008) as

$$WPC_{ij} = \frac{\sum_{k=1}^{n_j} \frac{Y_{ij(k)}}{c_{j(k)}}}{\sum_{k=1}^{n_j} \frac{1}{c_{j(k)}}} \tag{7}$$

where n_j is the number of bills in a congressional session authored by senator j, $c_{j(k)}$ is the number of cosponsors on senator j's k^{th} bill, where $k \in \{1, \ldots, n_j\}$, and $Y_{ij(k)}$ is a binary variable that is 1 if senator i cosponsored senator j's k^{th} bill, and is 0 otherwise. This measure ranges in value from 0 to 1, where $WPC_{ij} = 1$ if senator i is a cosponsor on every one of senator j's bills and $WPC_{ij} = 0$ if senator i is never a cosponsor any of senator j's bills.

Because we require binary edges for SAOMs, we focus only on very strong collaborations,

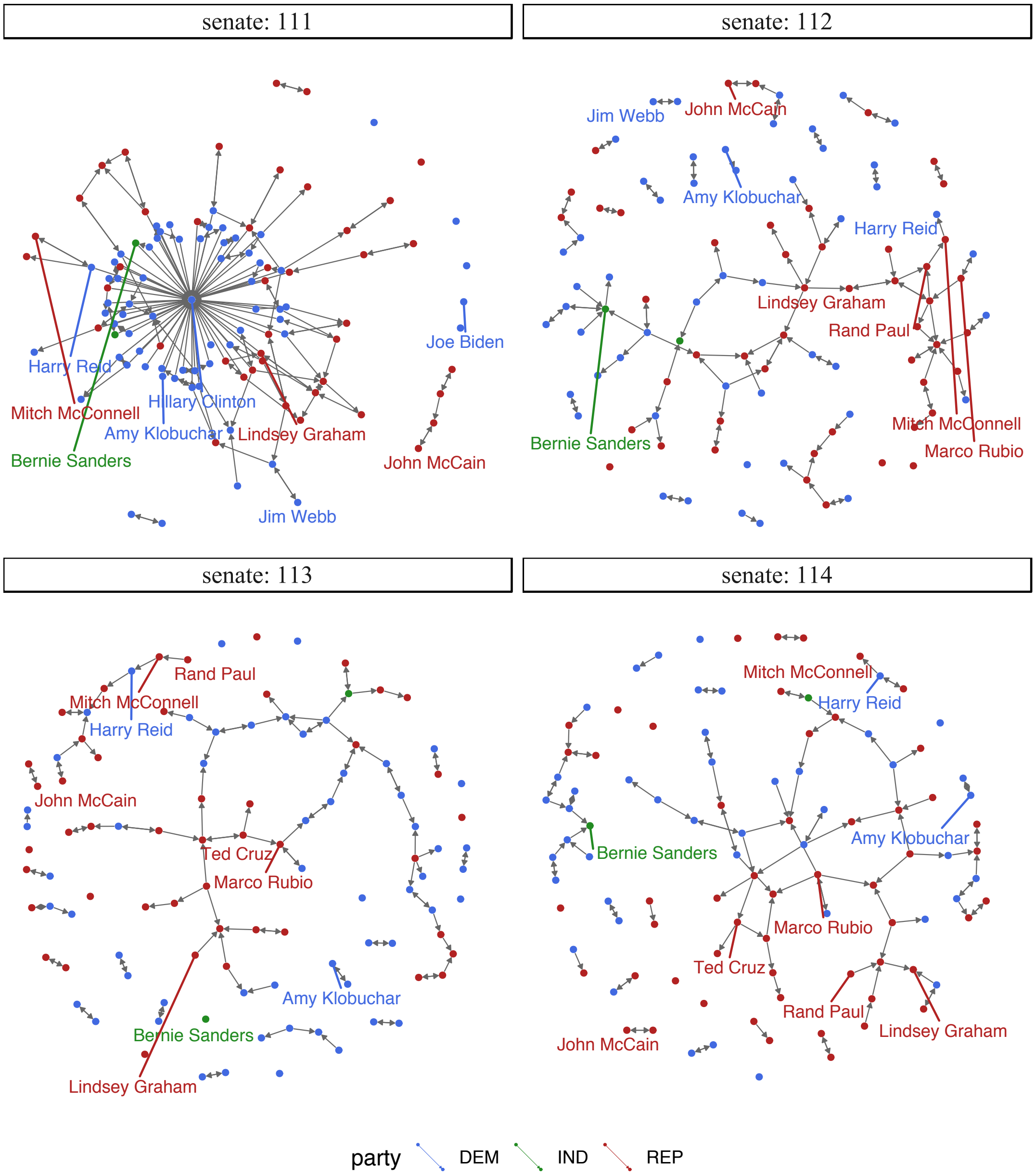


Figure 3: The collaboration network in the four senates during the Obama years, 2009-2016. Edges are shown only if the WPC between two senators is greater than 0.25. We use the FR layout algorithm here.

where WPC is high. For a senate collaboration network x, edges are defined as

$$x_{ij} = \begin{cases} 1 & \text{if } WPC_{ij} > 0.25 \\ 0 & \text{if } WPC_{ij} \leq 0.25. \end{cases} \tag{8}$$

The node-link diagram for the four senates during the Obama administration are shown in Figure 3. We can see that Senate 111 is much more densely connected than the other three, which have one large, sparsely connected component with a few smaller connected groups throughout. Some prominent names are shown to demonstrate their place in the network.

Models: To our example data, we fit three different SAOMs. Each SAOM uses a simple rate function, α_m, and an objective function with two or three parameters. The first model, M1, contains the absolute minimum number of parameters in the objective function $f_i(x)$:

$$f_i(x)^{M1} = \beta_1 s_{i1}(x) + \beta_2 s_{i2}(x), \tag{9}$$

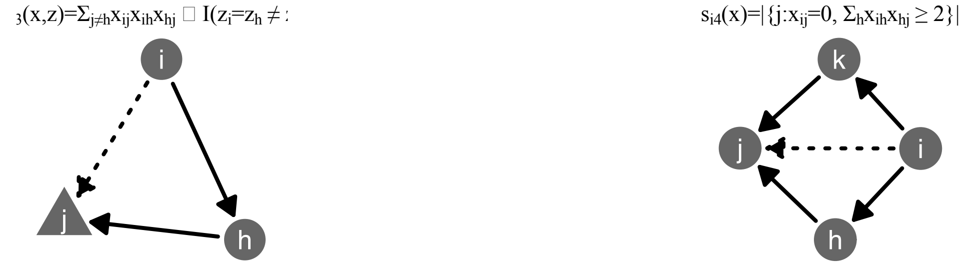
where s_{i1} is the density network statistic and s_{i2} is the reciprocity network statistic for actor i at the current network state x (see Table 1. The second and third models, M2 and M3, contain one additional parameter each in the objective function which were determined by a Wald-type test provided in the `RSiena` software to be significant, with p-values less than 0.05 (Ripley et al., 2016b). The M2 model contains an actor-level covariate parameter, and the M3 model contains an additional strutural effect in the objective function.

$$f_i(x,z)^{M2} = \beta_1 s_{i1}(x) + \beta_2 s_{i2}(x) + \beta_3 s_{i3}(x,z) \tag{10}$$

$$f_i(x)^{M3} = \beta_1 s_{i1}(x) + \beta_2 s_{i2}(x) + \beta_4 s_{i4}(x), \tag{11}$$

where $s_{i3}(x,z) = \sum_{j \neq h} x_{ij} x_{ih} x_{hj} \mathbb{I}(z_i = z_h \neq z_j)$, and $s_{i4}(x) = |\{j : x_{ij} = 0, \sum\limits_h x_{ih} x_{hj} \geq 2\}|$. These statistics are known as the *number of jumping transitive triplets* (JTT) and the *number of doubly achieved distances two effect* (N22), respectively. The first statistic emphasizes triad relationships formed between actors with different covariate values, while s_{i4} emphasizes indirect ties between actors. The covariate for the friendship data is the drinking behavior, and is the number of bills authored for the senate data. These additional effects are shown in Figure 4a and Figure 4b, where the ties affected by the parameter value are represented by dotted lines.

It is ofen difficult to know for certain how to interpret a fitted value of a parameter. We can make educated guesses based on the definition of the effect and the sign of the fitted value, but a direct interpretation is not always possible. By graphically exploring these


$_3(x,z)=\Sigma_{j\neq h}x_{ij}x_{ih}x_{hj}$ □ $I(z_i=z_h \neq$
i
j
h
$s_{i4}(x)=|\{j:x_{ij}=0, \Sigma_h x_{ih}x_{hj} \geq 2\}|$
k
j
i
h

(a) Realization of a JTT, where i is the ego, j is the alter, and h is the intermediary. The covariate is represented by the shape of the nodes.

(b) Realization of a N22 between actors i and j. The dotted tie is encouraged if β_4 has a positive value.

Figure 4: The additional network effects included in our models fit to the friends data. On the left, a JTT between i and j, who have different covariate values, and on the right, a N22 between i and j.

models, we aim to understand their effects, their interpretations, and the model fitting process better.

In this paper, we use three different model visualization techniques on SAOMs. First, Section 3 places the models in the data space. Then, Section 4 explores visualizations of various collections of models. Finally, in Section 5 we use model visualization to explore the underlying algorithms in the SAOM fitting process. We conclude with a discussion in Section 6.

3 Model in the data space

Every good data analysis includes both numerical and visual summaries of the data, so why restrict model description and diagnostics to numerical summaries? The concept of *model visualization* was developed to complement traditional model diagnostic tools such as R^2 values and residual plots (Wickham et al., 2015). In a modeling task, there are three different things called "model" under consideration: the model *family*, the model *form*, and the *fitted* model. The latter is usally what first comes to mind when considering a model, where a specified model is fit, and parameter estimates are reported.The *model form* describes the the model *before* the fitting process, defining which parameters are in the model within the context of the larger model family. Finally, the *model family* is the broadest description of the model. This is the type of model that you wish to fit to the data, and is chosen based on the problem, data, and knowledge at hand.

The model family, the model form, and the fitted model can each be visualized according to the three principles of model visualization: we can view the model in the data space, visualize collections of models, and explore the process of fitting the model, not just the end result. Since we have already decided on our model family, SAOMs, we shift our focus to the fitted model and the model form. Specifically, we want to learn more about how the *model form* we choose affects the *fitted model* by using our example data sets and our visualization toolbox. We use the data and models described in Section 2 to guide our visual explorations of SAOMs under the three principles of model visualization.

The first way we hope to better understand SAOMs is by viewing the model in the data space. In Wickham et al., they chose define the *data space* as "the region over which we can reliably make inferences, usually a hypercube containing the data" (Wickham et al., 2015, p. 206). But what does this definition mean for dynamic network models? We interpret the data space for our application as the combined data structures of the network: the node, edge, and time information.

The node, edge, and time data can be visualized together in various ways, and one tool that can help is the `R` package `geomnet` (Tyner and Hofmann, 2016). The network's node or edge data can be mapped to different visual features in the node-link diagram. The color, size, and shape of the points can be used to represent variables in the node data, while the color, linewidth, and linetype of the lines between points can be used to represent the edge variables. To view temporal changes, we can view the network at different timepoints side-by-side to see the evolution. Pulling all of this information together with `geomnet` allows the entire data space to be viewed at once.

To demonstrate, we use `geomnet` to visualize the connections in the 111^{th} United States Senate at two different points: when Hillary Clinton was in the senate, and after she left to become Secretary of State. Clinton was only in the 111^{th} senate from January 3-20, 2009, in the middle of her second term as the junior senator from New York. In that time, she authored two bills and was a cosponsor on 17s. There are many senators who

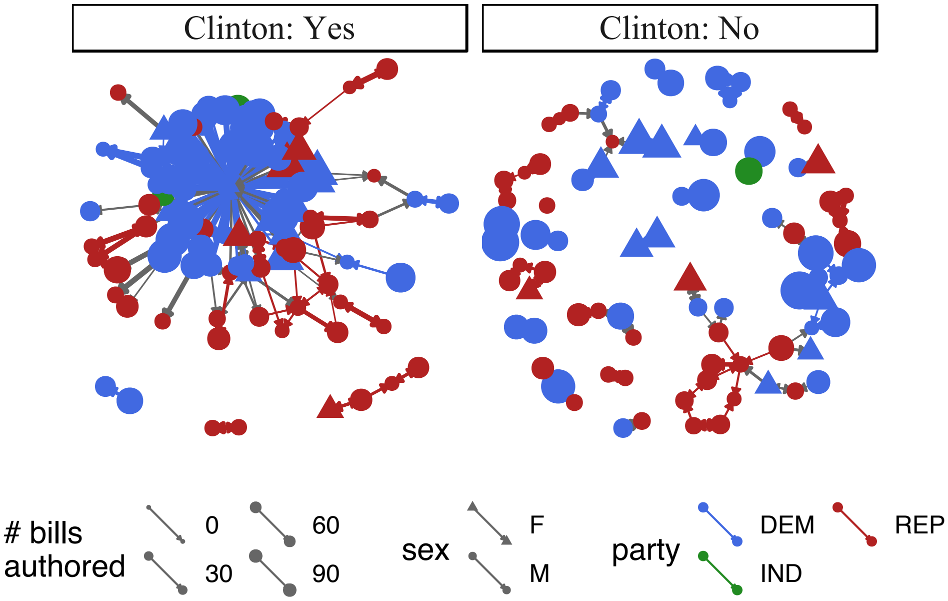


Figure 5: The 111th Senate at two times: while Clinton was in the senate in 2009 (on the left) and after she left (on the right). We map sex, party, and bills authored to the shape, color, and size of the nodes, respectively. We also map the WPC, to the edge linewidth.

cosponsored both of her bills, giving their edges to her a WPC of 1. So with Clinton included in the node-link diagram in Figure 5, the senate looks much more collaborative than it does without her. We then map our potential covariates to visual features: number of bills authored maps to the size of the node, gender maps to the shape, and party to the color. In addition, we visualize the strength of the tie by mapping the WPC value between the two senators to the linewidth of the edge. In this single visualization, we have viewed node information, edge information, and time.

Another way to view the model in the data space is through simulation from the model. No one network simulated from a SAOM is going to look like the data or represent the model, so we might want to look at an "average network" or "expected network" value. We frequently rely on averages and expected values in data analyses, but network models, especially those as complex as SAOMs, lack a expected network value measure. We could talk about expected values of parameters, but the parameters can be hard to interpret. There is no defined expected value of an simulation from a network model, though there are with most other model simulations. How then, can we arrive at an "average" network? We answer this question through visualization.

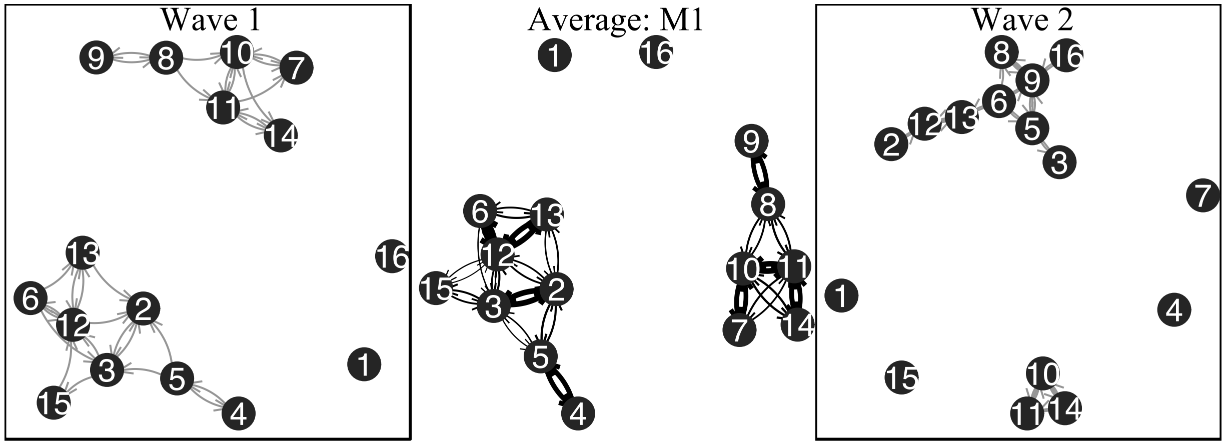


Figure 6: On the left, the first wave of observed data that is conditioned on in the model. On the right, the second wave of observed data. In the middle, a summary network from the first model fit to the data. This summary network represents 1,000 simulations of wave 2 using the values from the simple fitted model M1.

In Figure 6, we show an *average network* created with 1,000 simulations of the second wave of the network from Model 1. To make this average network, we first simulated 1,000 wave 2 and wave 3 observations of our small friendship example data from model M1, for which parameters had previously been estimated. We then aggregate the 1,000 simulated instances of wave 2, and by counting up the number of times each edge $i \to j$ appears in a network simulation. Then, we construct a single network for each time point with edge weights equal to the proportion of time the edge appears in our 1,000 simulations. This weighted network is shown in the middle panel of Figure 6, with edges present only in the average network if it appears in at least 51 of the 1,000 simulations. The edges that appear most frequently in the model are drawn with darker, thicker lines. On either side of the average network in Figure 6, we show the actual data, wave 1 on the left, and wave 2 on the right. We can see that the structure of the average network is much more similar to the first wave than to the second wave. However, the simulations are supposed to represent the second wave of data on the right of Figure 6. This is an indication that the simple model, M1, is doing a very poor job of capturing the change mechanism from the first to the second wave of observation. The average network can thus be used to help determine model goodness-of-fit. Because the the average network looks more like the first wave than the second wave, we can use the visualization in Figure 6 as evidence of poor model fit.

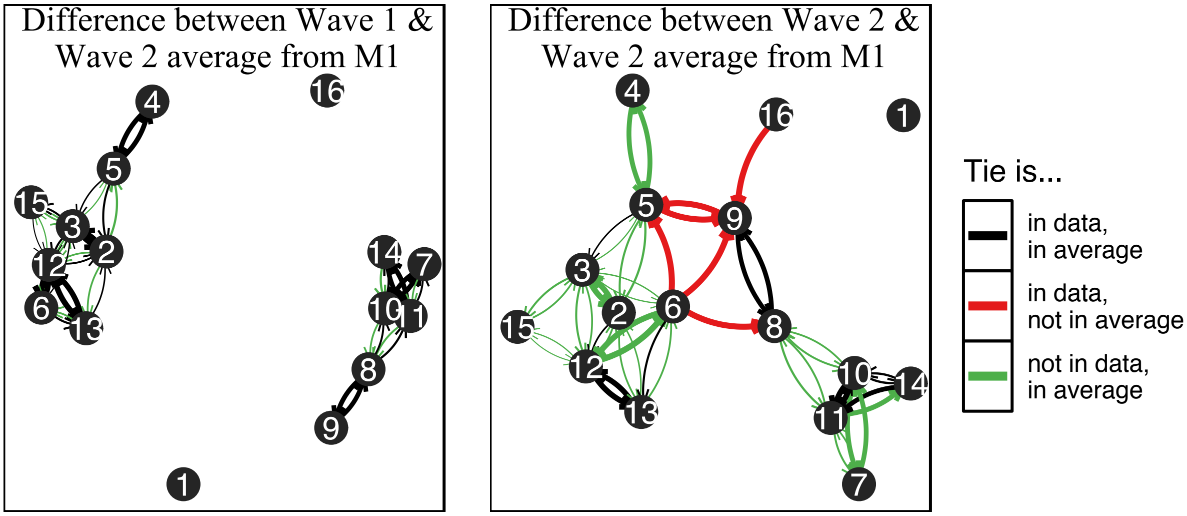

Difference between Wave 1 & Wave 2 average from M1
Difference between Wave 2 & Wave 2 average from M1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Tie is...
in data, in average
in data, not in average
not in data, in average

Figure 7: Network differences between the average network from M1 and Wave 1 and Wave 2 of the data. Some edges appear in both the data and the average Wave 2 from M1, while others only appear in the data or only in the average. Width of edges in the average network represent the proportion of appearances in the 1,000 simulations. The FR layout is used.

In Figure 7, we show *difference networks* between the average network from M1 and the first and second waves of data. These networks are constructed by computing the difference between two adjacency matrices. The lack of model fit is very obvious here, with the green edges representing ties that only show up in the average network, and the red edges representing ties in the data that the model misses completely. Again, the edgeweights demonstrate the frequency of the ties in the simulations from M1. On the left in Figure 7, we see the difference between the first wave of data and the average network. We only see green and black edges, meaning that the simulated networks consist primarily of edges in wave 1, plus some new edges that are not in the first wave of data. The additional green edges are all reciprocating black edges, meaning that the model predicts more reciprocity in ties than there actually is. On the right in Figure 7, we see the difference between the second wave of data and the average network. The model is simulating the second wave of data, so we would hope to see few differences between the average and wave2. However, there are six red edges, which are edges in wave 2 that do not appear at all in the average network, and there are several green edges that are in the average network, but not in the data. Although there are some edges that appear in both the average and wave 2, the majority of edges in the average are not in wave 2, and there are several missing edges. Thus, the model M1 is an incredibly poor fit, which we are able to see very easily by viewing the model in the data space in this way.

4 Collections of models

The second method of model visualization we use is visualizing collections of models. There are many possible ways to collect SAOMs together, so we chose four ways to create collections of models that would give us the most insight: (1) exploring the space of all possible models, (2) fitting the same model to the same data many times, (3) varying model settings, and (4) fitting the same model form to different data. We chose these four collections because they each explore something different about SAOMs. The first considers the many parameters to include in a SAOM, while the second results in different parameter estimates, so we can see how parameter estimates vary. The third collection shows how those many possible parameters affect the values of parameters in the fitted models. Finally, we explore how the same model behaves under two completely different circumstances.

4.1 Exploring the space of all possible models

The `RSiena` manual contains over eighty effects to include in a SAOM's objective function. In order to select parameters to include in the models for our example data, we searched through the possible effects available to model given the data structure to find *significant* effects. We tested for significance using the Wald-type tests built into `RSiena` for one-at-a-time effects testing. We start with the outdegree and reciprocity measures as the foundation of the models we fit, then add one effect, fit the model, test the additional effect for significance, and repeat for all possible parameters to add to the model. We performed this procedure, which we used to pick M2 and M3, for both the small friendship and the senate collaboration data. A visualization of the significant effects for the two example data sets are shown in Figure 13 in the Appendix.

4.2 Fitting the same model to the same data

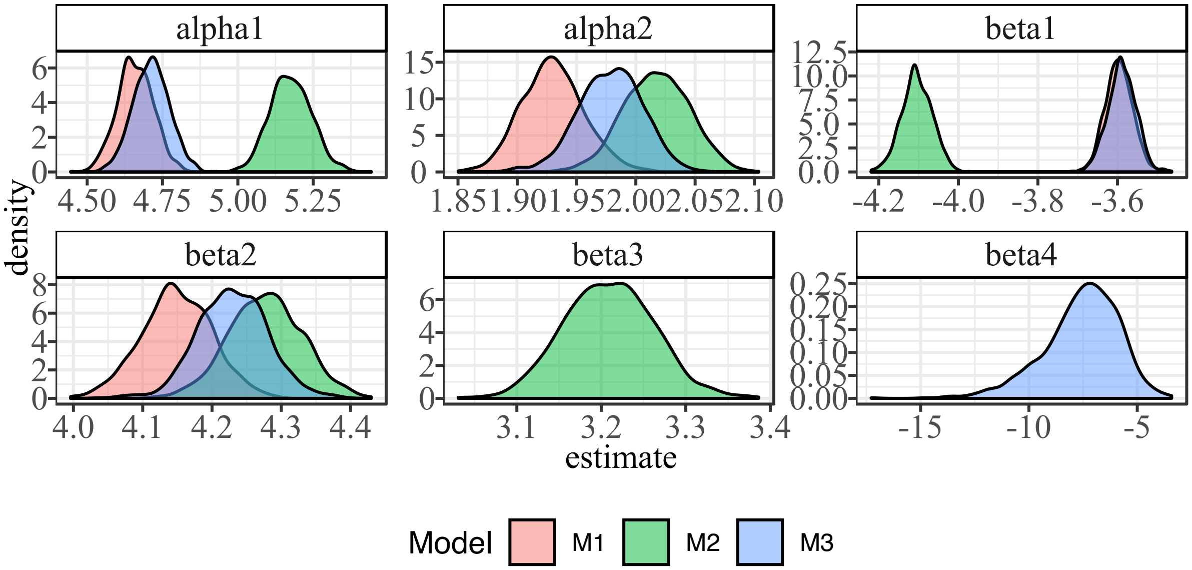


Figure 8: Distribution of fitted parameter values for our three SAOMs. The inclusion of β_3 or β_4 clearly has an effect on the distribution of the rate parameters, α_1 and α_2, and on the other objective function parameters β_1, β_2.

After fitting models M1, M2, and M3 to the friendship network 1,000 times each, we examine the distribution of the fitted values, which are shown in Figure 8. We can see from these distributions that the inclusion of the JTT parameter, β_3 is obviously affecting the distributions of the other four parameters included in all models, α_1, α_2, β_1, and β_2, while these parameters are less affected by the inclusion of β_4. When β_3 is included, its estimate is postitive, meaning that friendships between two girls with different drinking behaviors tend to form when there is an intermediary who is already friends with the two girls. The inclusion of this parameter leads to increases in the rate parameters' estimates, suggesting that the presence of the transitive triplet behavior means that the girls would also change friends more frequently. It is not clear, however, that the addition of a parameter to the objective function *should* effect the estimation of the rate parameters. The strong correlations between the estimates of the rate parameters and of the objective function parameters are shown in Figure 16 in the Appendix. In addition, the outegree parameter, β_1 decreases when β_3 is included, while the reciprocity parameter, β_2 increases. This implies the girls in the data prefer to form small, closely knit friend groups, as indicated by reciprocated ties and jumping transitive triplet formation, as opposed to being popular and having many friends. Thus, having many friends who do not reciprocate is more stronlgy discouraged by M2 than in the other models.

4.3 Varying model settings

The next method of viewing collections of models is with varying model settings. We have varied model settings already by choosing three different models M1, M2, M3 to fit to our example data sets. In Section 4.2, we fit these three models to the data 1,000 times, and examined the parameter densities. Here, we simulate from these three models. For simulation, we used the mean parameter values from the 1,000 fits of the corresponding model. We simulated 1,000 observations from each of the three models in this way. From these simulations, we created visualizations that represent an average network from models M1, M2, and M3 using the same method described in Section 3. These averages are shown in Figure 14 in the Appendix. We see that all models are failing to capture the structure of the second wave.

4.4 Fitting the same model to different data

In addition to fitting M1, M2, and M3 to the friendship data 1,000 times, we also fit them to the senate data 100 times. The means and standard deviations of the parameter estimates for each combination of model and data are given in Table 2, while the density plots of each parameter in the model for each data set is given in Figure 15.

Looking at Table 2 and Figure 15 in the Appendix, we see a few patterns in the estimates from both models. First, the table and the density plots demonstrate the same relationship between the outdegree parameter, β_1, and the reciprocity parameter, β_2. In both data sets and across all three models, the estimates of β_1 are all negative and hover between -5 and -3, while the estimates of β_2 are all positive and hover between four and five. This suggests that in both data sets, nodes are *encouraged* to form outgoing ties that are reciprocated and *discouraged* from forming ties that are not. Actors in these networks want reciprocated relationships: the students want to have close mutual friendships, and senators want mutual

	Friendship Data			Senate Data		
	M1	M2	M3	M1	M2	M3
α_1	4.660 (0.059)	5.176 (0.068)	4.712 (0.060)	3.344 (0.016)	3.349 (0.016)	3.340 (0.016)
α_2	1.930 (0.026)	2.017 (0.028)	1.979 (0.027)	2.480 (0.017)	2.487 (0.015)	2.483 (0.014)
α_3	–	–	–	2.221 (0.017)	2.227 (0.017)	2.224 (0.016)
β_1	-3.597 (0.033)	-4.104 (0.038)	-3.589 (0.035)	-4.979 (0.027)	-4.993 (0.025)	-4.987 (0.021)
β_2	4.149 (0.050)	4.277 (0.052)	4.230 (0.050)	4.954 (0.046)	4.974 (0.040)	4.970 (0.035)
β_3	–	3.209 (0.053)	–	–	-1.175 (0.789)	–
β_4	–	–	-7.582 (1.746)	–	–	-1.048 (0.486)

Table 2: The means (std. dev.) of parameter values estimated from repeated fittings of $M1, M2, M3$ to the two example data sets.

support for their bills.

The inclusion of β_3 for the friendship data had a noticeable effect on the other parameters in the model. This is not true in the senate data. Looking at the estimates of β_4, we see that the estimates for the senate data are nearer to zero, suggesting that this effect, which considers indirect ties, is not important for the senate data. This effect is, however, very strong in the friendship data with a mean of about -7.5, suggesting that indirect ties are discouraged from forming: teenage girls want to have direct friendships. This indirect relationship variable does not help explain the senate collaboration structure, but it does help clarify the teenage friendship structure.

5 Explore algorithms, not just end result

The last principle of model visualization we use is exploring the process of fitting the model, instead of just focusing on the end result. This principle is perhaps the most important for SAOMs because the model fitting process in `RSiena` involves several simulation steps that are hidden from the user. Hiding the MCMC steps is practical and efficient if the goal is to fit a model to a set of longitudinal network data, obtain parameter estimates, and draw conclusions or make predictions. We are more interested in *how* the models are fit, so we extracted and explored the different steps of that process instead of allowing them stay hidden.

One of the hidden steps in the SIENA method of moments algorithm is the "microstep" process. A series of microsteps is obtained by simulating from the model in its current state, $x(t_m)$ with current parameter values $\theta_0 = \{\alpha_{1_0}, \ldots \alpha_{(m-1)_0}, \beta_{1_0}, \ldots, \beta_{K_0}\}$, to the next state, $x(t_{m+1})$. One tie changes at a time until the simulated network has achieved the same number of differences, C, from $x(t_m)$ as $x(t_{m+1})$, where C is definined in Equation 3.

This simulation process follows the steps of the continuous-time Markov chain. Each tie change in the CTMC is referred to as one "microstep". At each microstep, an "ego node" is selected to make a change, and the chosen ego node randomly makes one change in its ties according to the probabilities, $\{p_{ij} : i \neq j \in \{1, \ldots, n\}\}$ defined in Equation 5, determined by its objective function. The options for change are removing a current tie, adding a new tie, or making no change at all. Between two network observations $x(t_m)$ and $x(t_{m+1})$, there can be dozens, hundreds, or even thousands of microsteps, depending on the size of the network and the number of tie changes between two network observations. We want to simulate and view these in-between steps in order to better understand the behavior of the underlying continuous-time Markov chain.

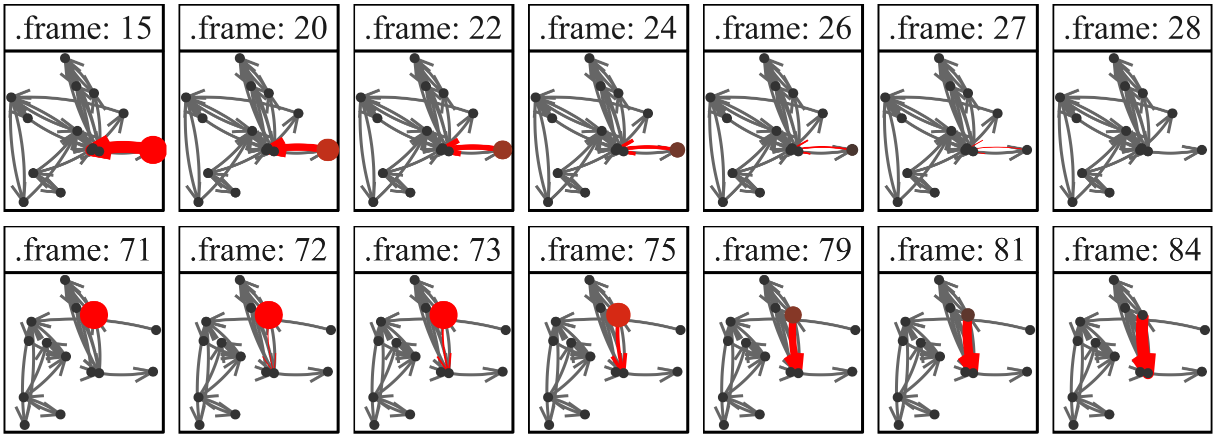


Figure 9: A selection of images in the microstep animation. The selected edges and nodes are emphasized by changing the size and the color, then when edges are removed, they fade out, shrinking in size, while the nodes change color and shrink to blend in with the other nodes. Frames 15-28 show a an edge disappearing, while frames 71-84 show an edge appearing.

The first vizualization we present here is an animation of the simulated microsteps, when fitting model M1, that form the transition steps of the CTMC from wave 1 to wave 2 of the small friendship network example shown in Figure 2. Movies similar to this animation were used to visualize the changes of dynamic networks in Moody et al. (2005). When each ego node is selected in a microstep, it is emphasized in the animation, then the associated edge either appears or disappears. If there are no changes at a particular microstep, no changes are seen. Some frames of the animations are shown in Figure 9, and the full movie, created with `tweenr` and `gganimate` can be viewed at `https://vimeo.com/240089108` (Pedersen, 2016; Robinson, 2016).

The top row of Figure 9 shows an edge being removed, and the bottom row shows one being added. In both cases, the ego node is emphasized with color and size changes. When an edge is removed, the edge that currently exists is emphasized with the same color and size change as the node. In the animation, the edge shrinks and disappears as the ego node shrinks and changes color back to its original black. If an edge is being added, the red edge appears from nothing, grows large for emphasis, then changes color and size to match the rest of the edges.

In the network animation, we see the modeled steps of the unobserved CTMC process between two obsered networks. We see each part of the SAOM come into play: we first see the rate at which the nodes are selected to change, then we see the result of the actor maximizing its objective function by either deleting or adding a node. In addition, the layout of the nodes changes as edges are removed or added, which gives us a better sense of how the overall network structure changes with these individual tie changes.

We continue to use animation to view the changing structure of the network via adjacency matrix visualizations. Since there are 16 nodes in the data, numbered 1-16, we first use that order on the x and y axes for the matrix visualization. Viewing the adjacency matrices with this arbitrary ordering does not provide much information to the viewer about the underlying structure of the network, as shown in Figure **??**. This lack of perceived structure would contribute to further confusion as the network chages, so we adjust the ordering so that the viewer can better perceive the structure of the network. This process is known as matrix seriation (Liiv, 2010).

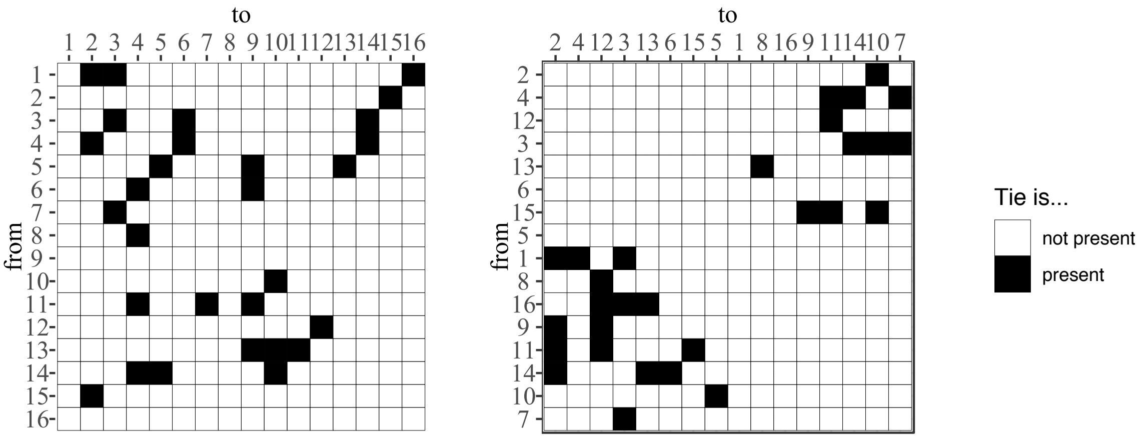


Figure 10: On the left, the starting friendship network represented in adjacency matrix form, ordered by vertex id. On the right, the same adjacency matrix is presented after ordering the vertices by one repetition of the microstep simulation process from wave one to wave two.

To reorder the vertices for the matrix visualization, we first constructed a cumulative adjaceny matrix, $\mathbf{A}^{cum}$, for the series of microsteps simulating the network from $x(t_m)$ to $x(t_{m+1})$. A single entry in the cumulative adjacency matrix, $\mathbf{A}_{ij}^{cum}$, is the total number of times the edge $i \to j$ appears in the network from the initial observation, $x(t_m) \equiv X(0)$ to the final result of the last microstep, $X(R)$, where R is the total number of microsteps taken:

$$\mathbf{A}_{ij}^{cum} = \sum_{r=0}^{R} X_{ij}(r). \tag{12}$$

We then performed a principal component analysis (PCA) on $\mathbf{A}^{cum}$, and used the values of the first principal component to order the vertices on the x and y axes for the adjacency matrix animation. In Figure **??**, we present the first wave of data ordered by the arbitrary node ID order alongside the seriated adjacency matrix ordered by the first principal component loading on the cumulative adjacency matrix, $\mathbf{A}^{cum}$. Using PCA on $\mathbf{A}^{cum}$ to order the rows and columns of the adjacency matrix visualization clearly shows the two distinct connected components in the first wave of the network, which are difficult to find in the arbitrarily ordered visualization.

We also use the PCA seriated layout to fix the layout in the animation of one of the microstep process simulations, some frames of which are shown in Figure 11. This animation is very simple: a square appears or disappears in the animation as that edge appears or disappears in the microstep process. Through this animation, which can be viewed at `https://vimeo.com/240092677`, we can see edges appearing, and then later on disappearing. These in-between steps are not shown when we look at the network at our discrete observation points, so by viewing this animation we can gain a better understanding of the underlying dynamics of this model.

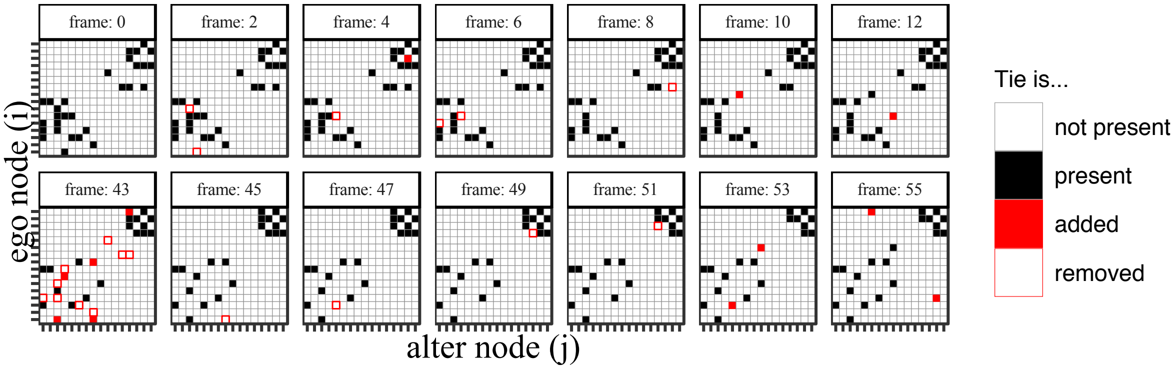


Figure 11: A selection of frames from the adjacency matrix visualization animation for one series of microsteps. (Ego and alter node labels are removed to declutter the figure.) Starting at frame 0, there are two clearly connected components: one in the bottom left corner, and one in the top right corner. By the end, the component in the bottom left has spread out and become sparsely connected, while the top right component has shrunk, but remains closely connected.

We also attempt to better understand the microstep process by visualizing the empirical transition probabilites for the first microstep in the process. These probabilities are defined in Equation 5. We only do the first step of many because the `RSiena` transition probabilities for any two edges i, j after the first step are only directly comparable for identical microsteps due to the conditioning on the current network state in the model. Thus we have 1,000 transition probabilities to examine: one transition probability for the first microstep taken for each of the simulations.

In Figure 12, we build on the concept of the ordered adjacency matrix of Figure 10. This adjacency matrix visualization shows the transition probabilities of all ties that are changed in the first microstep of the 1,000 simulations. The figure is noticeably sparse: of the $16 \cdot 15 = 240$ possible changes for the CTMC to make, only 103, or about 43%, are taken

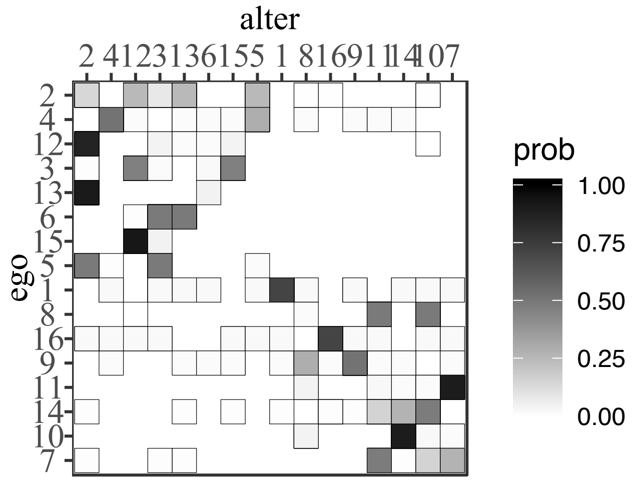


Figure 12: An adjacency matrix visualization showing the empirical transition probabilities for the first microstep in 1,000 simulations. The ego node is on the y-axis, and the alter node is on the x-axis. There were many ties with empirical probability zero.

in the 1,000 simulated chains. This leaves a very large area of our possible network space unexplored by the first steps in the SAOM model fitting process. We present additional evidence for lack of coverage in Figure 17 the Appendix. The Figures 12 and 17 show that the model M1 and the SAOM fitting process do not explore the data space enough to adequately capture the network change mechanism, a finding we were only able to discern thanks to model visualization techniques.

6 Discussion

We have used novel visualization methods in order to better understand the family of models known as stochastic actor-oriented models for social network data. By viewing the model in the data space, visualizing collections of models, and looking at the underlying algorithms, we have been able to gain knowledge and appreciation for these complicated models and everything that goes into them. When we looked at the model in the data space by viewing the average network simulated from a model, we were able to see the lack of model fit. When visualizing collections of models, we uncovered concerning relationships between parameters. Finally, when we explore the algorithms, we gain a better understanding of the model and its behavior.

We have only just begun to scratch the surface of these complicated and multi-layered models for social networks. The `RSiena` software is incredibly powerful, and can fit a whole slew of much more flexible stochastic actor-oriented models than we have examined here. If, for example, a researcher thinks the network structure or an actor covariate effects the rate of change of the network, there is a way to incorporate that belief into the rate function of the SAOM. In addition, more than one actor-level covariate can be included in the model, and many more than three parameters can be included in the objective function itself. The flexibility of `RSiena` allows the user to specify which parameters lead to tie creation, and which parameters lead to tie dissolution. We have used "evaluation" parameters, which are agnostic, allowing the data to decide which parameters affect creation or dissolution via the sign (Ripley et al., 2017). Finally, SAOMs and `RSiena` are able to also model behavior change of the actors in the network, which again is a capability we did not explore here.

Appendix: Additional visualizations

Section 4.1

In Figure 13, we see for the friendship data and the senate data that most of the significant effects have absolute value less than ten. In addition, the p-values for the effects from the friendship data are more spread out than the p-values for the senate data, which are concentrated at about 0.02 or less. This may suggest that larger data sets tend to result generally in smaller p-values, just like a larger sample size results in smaller p-values in a t-test. Since the tests are Wald-type tests, this makes sense.

Section 4.2

To create the average network visualization shown in Figure 14, we follow the same procedure as in Section 3, counting occurrences of each possible edge in the simulations, resulting in a summary network with weighted edges representing the number of times an edge appeared in the simulated wave 2 when simulating from the SAOM 1,000 times. As in Figure 6, edges only appear in the average networks if they appear more than 5% of the time in the simulations. In Figure 14, we show the "average" network from the three mod-

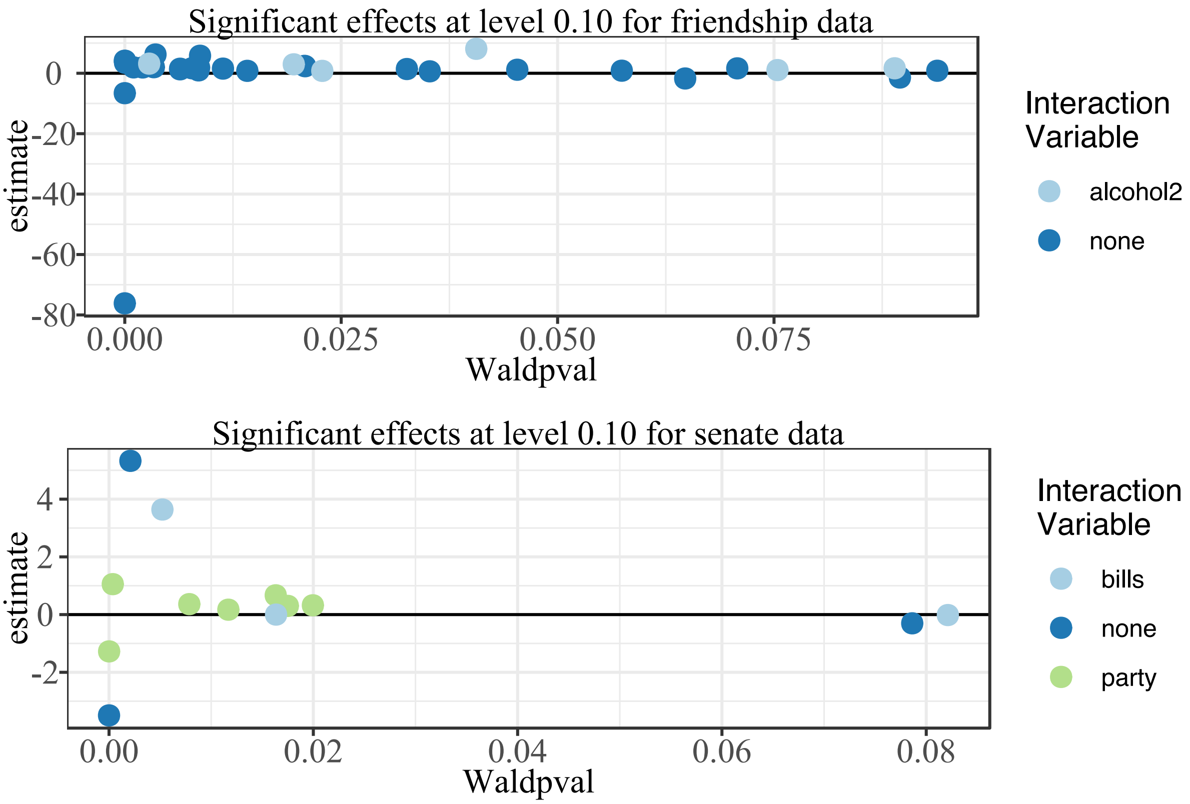


Figure 13: From Section 4.1. Significant effects for the two data sets, at a significance level of 0.10 or lower as calculated by the Wald-type test available in the SIENA software.

els we fit and the first and second waves of data. Comparing the three averages to waves 1 and 2, we see that they have very similar structure to wave 1. Model 2, which included the transitive triplet parameter, seems to have created a larger connected component overall than models 1 and 3. In particular, if we look at the group of nodes $\{10, 11, 14\}$, we see they are very strongly connected within the three average networks, and they are completely separate from the other nodes in the true wave 2. None of the three average networks show node 16 gaining ties as it does in wave two, nor do they show nodes 4 and 7 becoming isolated. In Model 2, however, the ties to node 7 appear much weaker than in Model 1 or Model 2, suggesting that of the three, Model 2 may be the best fit for our data.

Section 4.3

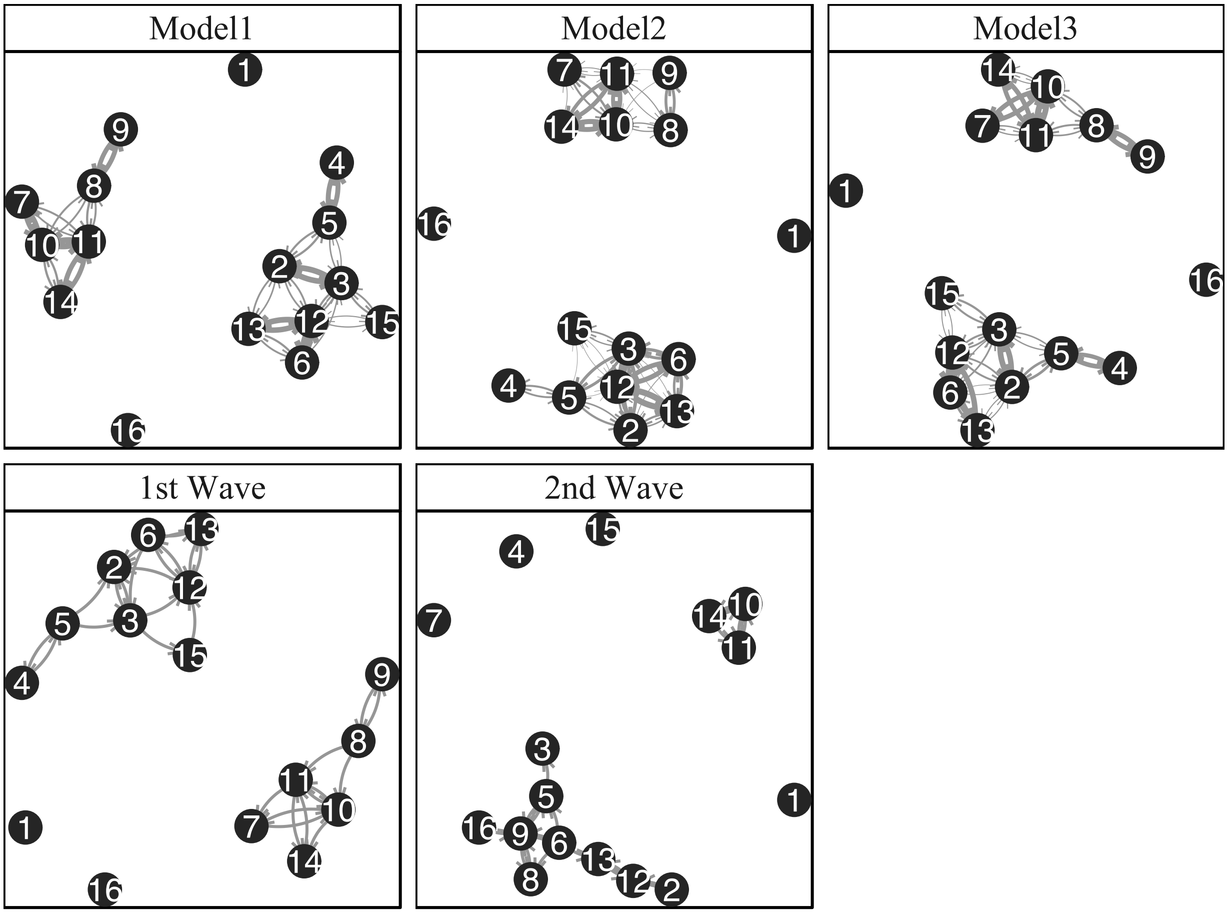


Figure 14: The node-link diagrams from the three "average" networks that we calculated are in the top row, and the true wave 1 and wave 2 data are shown in the bottom row above. There is some difference between the three models, but overall, these three models cannot capture the structure in the true second wave of data.

The density plots in Figure 15 show the distribution of the parameter estimates for the objective function parameters in M1, M2, and M3 the two example data sets we used. The most notable difference is in the values of β_1, β_2. In the senate data, the inclusion of β_3 has no effect on the estimates of β_1, β_2, while β_3 does seem to have an effect with the friendship data.

Section 4.4

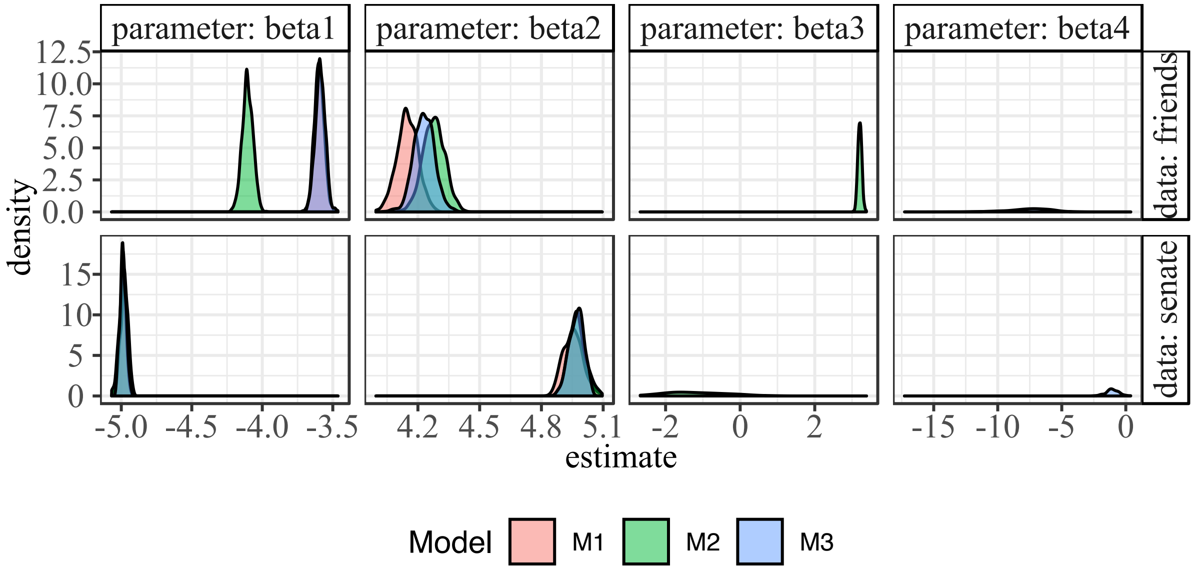


Figure 15: Density plots of objective function parameter estimates from repeatedly fitting models M1, M2, and M3 to the example data. Outdegree and reciprocity have the same inverse relationship for both data sets. For the friendship data, the inclusion of β_3 has strong effect on the estimates of the other two parameters, but not for the senate data.

In Figure 16, we examine correlations between each of pair of parameters within each model and overall. The strongest correlation within each model is between β_1 and β_2, with absolute value of correlation between those two parameter values greater than 0.90 in all three models. The β_1 parameter is also highly correlated with the β_3 parameter within model M2, but it is not as highly correlated with the β_4 parameter in model M3. It might therefore be advisable to consider only models that either allow β_1, or β_2. Looking at the high correlation with α, we might switch to a model without β_1.

Section 5

Finally, we combine 1,000 simulations from model M1 into a visualization that displays the entire microstep process for all simulations in Figure 17. To make this visualization, we first assign each possible edge an edge ID number so that we can keep track of it throughout all the microsteps and all the simlations. Then, we count up the total number of times each edge appears in the microstep process in each of the 1,000 simulations for use as an ordering

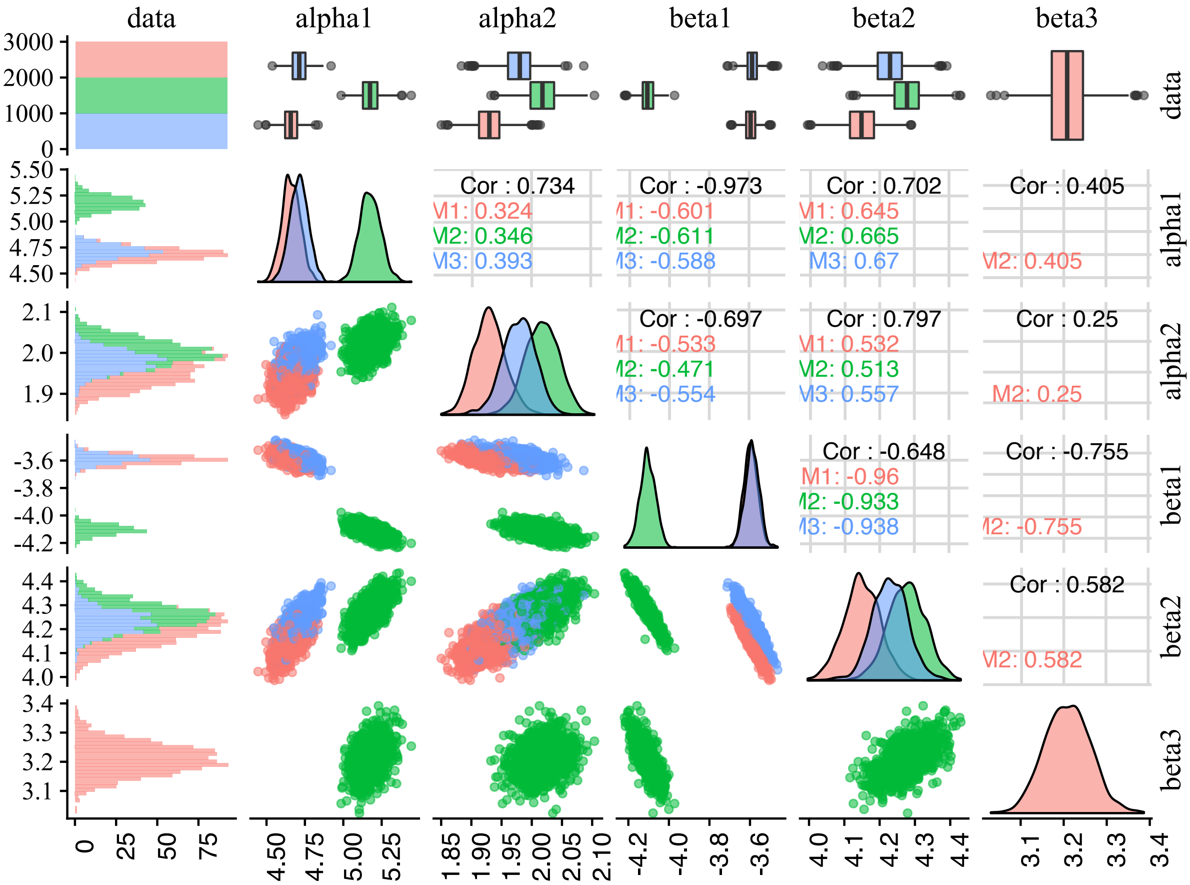


Figure 16: A matrix of plots demonstrating the strong correlations between parameter estimate in our SAOMs. The strongest correlation within each model is between β_1 and β_2.

variable later. We also count up the number of times an edge occurs in each microstep number in the 1,000 simulations. Since the number of microsteps in the process varies, the number of times an edge occurs decreases as the microstep number increases. Next, we compute a proportion, which we call the occurrence percentage, which is the number of times the edge occurred in a microstep divided by 1,000. Finally, we visualize all this information together in Figure 17. In this plot, all possible edges are shown, and we see that every one of the $16 \times 15 = 240$ possible edges in the network occurs at some point in the simulation process. We also see, however, that the process struggles to focus in on the edges in the second wave of the data. Ideally, we would like to see more occurrences of the edges which appear in the second wave of data. But, about half of the edges in wave two are in the bottom half when ordered by number of occurrences, meaning they do not appear as much as they would if the model was truly excellent at capturing the mechanisms of tie change in the network. This solidifies what we found in Figure 12: the model M1 and the SAOM fitting process do not explore the data space enough to adequately capture the network change mechanism.

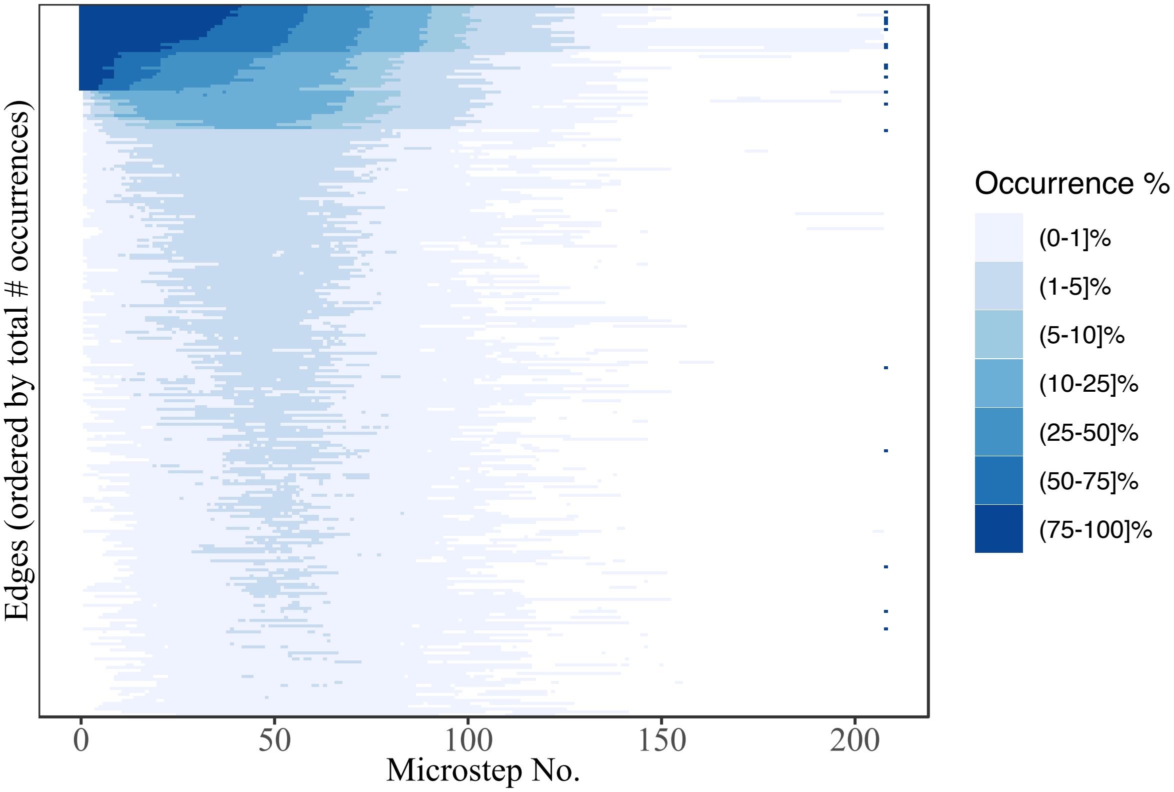


Figure 17: Visualizing all microsteps taken in 1,000 simulations from the model M1. The occurrence percent is split up into groups to correspond with its distribution: only about 10% of the possible edges appear more than 10% of the time in the 1,000 simulations, while about 60% appear less than 1% of the time. The first wave network is shown at microstep 0, and the second wave of the network is shown as the last microstep for comparison. We see that it is rare for a microstep process to last longer than 150 steps, and also that the edges that appear past the 150th step tend to be in either the first wave or the second wave.

Much of the code used to create these plots has been packaged an posted online at https://github.com/sctyner/netvizinf and https://github.com/sctyner/geomnet (also on CRAN). Additional, non-packaged code can be found at https://github.com/sctyner/SAOM-removing-blindfold.

References

Fekete, J.-D. (2009), "Visualizing Networks using Adjacency Matrices: Progresses and Challenges," in *11th IEEE International Conference on Computer-Aided Design and Computer Graphics*, pp. 636– 638.

Frank, O. and Strauss, D. (1986), "Markov Graphs," *Journal of the American Statistical Association*, 832–842.

Fruchterman, T. M. and Reingold, E. M. (1991), "Graph Drawing by Force-Directed Placement," *Software: Practice and Experience*, 21, 1129–1164.

Ghoniem, M., Fekete, J.-D., and Castagliola, P. (2005), "On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis," *Information Visualization*, 4, 114, copyright - Palgrave Macmillan Ltd 2005; Document feature - charts; graphs; tables; references; equations; Last updated - 2012-01-28.

Gibson, H., Faith, J., and Vickers, P. (2013), "A survey of two-dimensional graph layout techniques for information visualisation," *Information Visualization*, 12, 324–357, copyright - SAGE Publications Jul 2013; Document feature - Tables; ; Last updated - 2013-08-05.

Gross, J. H., Kirkland, J. H., and Shalizi, C. R. (2008), "Cosponsorship in the U.S. Senate: A Multilevel Two-Mode Approach to Detecting Subtle Social Predictors of Legislative Support," *Unpublished Manuscript*.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002), "Latent Space Approaches to Social Network Analysis," *Journal of the American Statistical Association*, 97, 1090–98.

Kamada, T. and Kawai, S. (1989), "An Algorithm for Drawing General Undirected Graphs," *Information Processing Letters*, 31, 7–15.

Knuth, D. E. (2013), *Combinatorics: Ancient and Modern*, Oxford University Press, chap. Two thousand years of combinatorics.

Liiv, I. (2010), "Seriation and Matrix Reordering Methods: An Historical Overview," *Statistical Analysis and Data Mining*, 3, 70–91.

Michell, L. and Amos, A. (1997), "Girls, pecking order and smoking," *Social Science & Medicine*, 44, 1861 – 1869.

Moody, J., McFarland, D., and Bender-deMoll, S. (2005), "Dynamic Network Visualization," *American Journal of Sociology*, 110, 12061241.

Pedersen, T. L. (2016), *tweenr: Interpolate Data for Smooth Animations*, r package version 0.1.5.

Ripley, R., Boitmanis, K., Snijders, T. A., and Schoenenberger, F. (2016a), *RSiena: Siena - Simulation Investigation for Empirical Network Analysis*, R package version 1.1-304/r304.

— (2016b), *RSienaTest: Siena - Simulation Investigation for Empirical Network Analysis*, R package version 1.1-294.

Ripley, R. M., Snijders, T. A., Boda, Z., Vrs, A., and Preciado, P. (2017), *Manual for RSiena*, University of Oxford: Department of Statistics; Nuffield College; University of Groningen: Department of Sociology, `https://www.stats.ox.ac.uk/~snijders/siena/RSiena_Manual.pdf`.

Robbins, H. and Monro, S. (1951), "A stochastic approximation method," *The Annals of Mathematical Statistics*, 22, 400–407.

Robinson, D. (2016), *gganimate: Create easy animations with ggplot2*, r package version 0.1.0.9000.

Schloerke, B., Crowley, J., Cook, D., Hofmann, H., Wickham, H., Briatte, F., Marbach, M., and Thoen, E. (2016), *GGally: Extension to ggplot2.*, R package version 1.3.0.

Snijders, T. A. (2016), *Siena Algorithms*, University of Oxford: Department of Statistics, `https://www.stats.ox.ac.uk/~snijders/siena/Siena_algorithms.pdf`.

Snijders, T. A. B. (1996), “Stochastic actor-oriented models for network change,” *Journal of Mathematical Sociology*, 21, 149–172.

— (2001), “The Statistical Evaluation of Social Network Dynamics,” *Sociological Methodology*, 31, 361–395.

Tyner, S. and Hofmann, H. (2016), *geomnet: Network Visualization in the 'ggplot2' Framework*, r package version 0.2.0.9001.

Wickham, H., Cook, D., and Hofmann, H. (2015), “Visualizing statistical models: Removing the blindfold,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8, 203–225.

Yin, G. G. and Zhang, Q. (2010), *Continuous-Time Markov Chains and Applications: A Two-Time-Scale Approach*, New York: Springer, 2nd ed., iSBN 978-1-4614-4345-2.