

Signed Network Modeling Based on Structural Balance Theory

Tyler Derr*

Charu Aggarwal†

Jiliang Tang*

Abstract

The modeling of networks, specifically generative models, have been shown to provide a plethora of information about the underlying network structures, as well as many other benefits behind their construction. Recently there has been a considerable increase in interest for the better understanding and modeling of networks, but the vast majority of this work has been for unsigned networks. However, many networks can have positive and negative links (or signed networks), especially in online social media, and they inherently have properties not found in unsigned networks due to the added complexity. Specifically, the positive to negative link ratio and the distribution of signed triangles in the networks are properties that are unique to signed networks and would need to be explicitly modeled. This is because their underlying dynamics are not random, but controlled by social theories, such as Structural Balance Theory, which loosely states that users in social networks will prefer triadic relations that involve less tension. Therefore, we propose a model based on Structural Balance Theory and the unsigned Transitive Chung-Lu model for the modeling of signed networks. Our model introduces two parameters that are able to help maintain the positive link ratio and proportion of balanced triangles. Empirical experiments on three real-world signed networks demonstrate the importance of designing models specific to signed networks based on social theories to obtain better performance in maintaining signed network properties while generating synthetic networks.

1 Introduction

Network modeling has been shown to provide a plethora of information about the underlying network structures, as well as many other benefits behind their construction. It aims to design a model to represent a complex network through a few relatively simple set of equations and/or procedures such that, when provided a network as input, the model can learn a set of parameters to

construct another network that is as similar to the input as possible. Ideally this would result in many observable/measurable properties being maintained from the input to the generated output network. In unsigned networks, the typical desired and modeled properties are the power law degree distribution [3, 6, 5, 1], assortativity [10], clustering coefficients [11], and small diameter [6].

Nowadays, more data can be represented as large networks in many real-world applications such as the Web [14], biology [16], and social media [9]. Increasing attention has been attracted in better understanding and modeling networks. Traditionally network modeling has focused on unsigned networks to preserve the aforementioned properties. However, many networks can have positive and negative links (or signed networks [2, 4]), especially in online social media, but this then raises the question – whether dedicated efforts are needed to model signed networks in addition to the unsigned techniques.

Signed networks are unique from unsigned not only due to the increased complexity added to the network by having a sign associated with every edge, but also (and more importantly) because there are social theories, such as balance theory, that play a key role driving the dynamics and construction of the networks. For example, in unsigned networks we have the property of transitivity and we see a large amount of local clustering (i.e., formation of triangles). In comparison, with signed networks, not only are their patterns in the network driving local clustering, but also in the distribution of triangles found in the network. Suggested by balance theory [4], some triangles are more likely to be formed (i.e., balanced) than others (i.e., unbalanced) in signed networks. Hence, modeling signed networks requires to preserve not only the ratio of positive and negative links, but also other properties such as the distribution of formed triangles. However, these mechanisms are not incorporated into unsigned network modeling. Thus, there is a need to specifically design network models for signed networks to correctly capture their properties.

In this paper, we aim to design a network model

*Data Science and Engineering Lab at Michigan State University. Email: derrtyl, tangjili@msu.edu.

†IBM T.J. Watson Research Center. Email: charu@us.ibm.com.

specific to signed networks. We propose a model for signed networks based on the Chung-Lu model, which targets to preserve three key properties of signed networks – (1) degree distribution; (2) positive/negative link ratio and (3) proportion of balance/unbalanced triangles suggested by balance theory. In particular, it introduces a triangle balancing parameter and a sign balancing parameter to control the distribution of formed triangles and signed links, respectively. Our contributions are summarized as follows:

- Provide an automated approach to estimate a triangle balancing parameter and a sign balancing parameter from the input signed network;
- Propose a balanced signed Chung-Lu model (BSCL), which can preserve three key properties of signed networks simultaneously; and
- Conduct experiments on three real-world signed networks to demonstrate the effectiveness of the proposed framework BSCL in preserving the signed network properties.

The rest of this paper is organized as follows. In Section 2, we review related work in network modeling with the focus on signed networks. Next, in Section 3, we formally introduce the generative signed network modeling problem. Then we motivate the need for dedicated signed network models before presenting our proposed balanced signed Chung-Lu model (BSCL), which is based upon structural balance theory in Section 4. This includes both the network generation process, parameter learning method, and time complexity analysis of both stages of our model. Section 5 presents the experiments we have performed to evaluate our model against existing methods and other baselines as well as examining the performance of our parameter learning algorithm. Conclusions and future work are given in Section 6.

2 Related Work

In this section, we briefly review the work that have attempted to model signed networks. However, there is no existing work focused on signed network generative modeling. This means they do not provide a mechanism for learning parameters to construct similar signed networks to a given input network, but instead parameters would need to be hand-picked.

In [8] Ludwig and Abell proposed an evolutionary model for signed networks that also made use of two parameters. One of the parameters is defined as the “friendliness” index and controls the probability a randomly created edge will be positive or negative. The

other parameter is a threshold that represents the maximum amount of unbalanced triangles each user of the network can participate in. The primary downside of this method is that it is not a generative model (i.e., parameters must be hand-picked and cannot be learned). A more recent model was based on a graph-theoretic approach to model the formation of signed networks [9]. In this method each user tries to minimize their social stress by attempting to individually follow balance theory to maximize the number of balanced minus unbalanced triangles they are participating in. This work is different in that they focused more on the theoretical analysis of balance theory in signed networks and have the objective of taking an existing network and optimizing the edges by either removing, flipping signs, or including new edges, such that the network becomes balanced (i.e., all triangles in the network are balanced). This work is not comparable to ours in that they do not attempt to generate synthetic networks with similar properties, but instead focus on adjusting/evolving existing networks to build complete networks that adhere to balance theory. In [17], an interaction-based model is presented to construct signed networks with a focus on preserving balanced triangles in the network. Their model uses an ant-based algorithm where pheromone is placed on edges where a positive (negative) amount represents a positive(negative) edge with the magnitude of the pheromone being the strength of the edge. During each step of the algorithm, for each node, a local and global process is performed. The global step allows for longer distance connections to be made proportional to the degree of the nodes. In comparison, the local step is used for reinforcing the pheromone on a triangle in the network and adjusts the third edge to adhere to balance theory based on the first two edges in the triangle. Note that this method contains 6 parameters and none can be learned by their model, but instead, require them to be hand-crafted.

3 Problem Statement

A signed network \mathcal{G} that is composed of a set of N vertices $V = \{v_1, v_2, \dots, v_N\}$, a set of M^+ positive links \mathcal{E}^+ and a set of M^- negative links \mathcal{E}^- . Let $\mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-$ represent the set of $M = M^+ + M^-$ edges in the signed network when not considering the sign. Note that in this work, we focus on undirected signed networks and would like to leave modeling directed signed networks as one future work.

We can formally define the generative signed network modeling problem as follows:

Given a signed network $\mathcal{G}_I = (V_I, \mathcal{E}_I^+, \mathcal{E}_I^-)$ as input, we seek to learn a set of parameters Θ for a given model \mathcal{M} that can retain the network properties found in \mathcal{G}_I ,

such that we can construct synthetic output networks $\mathcal{G}_o = (V_o, \mathcal{E}_o^+, \mathcal{E}_o^-)$, using \mathcal{M} based on Θ , that closely resemble the input network in terms of measured network properties.

4 The Proposed Signed Network Model

Traditionally network modeling has focused on unsigned networks and preserving unsigned network properties such as the power law degree distribution [3, 1, 6, 5], assortativity [10], clustering coefficients [11], and small diameter [6]. Although a previous study demonstrated that degree of signed networks also follow power law distributions [15], signed networks have distinct properties from unsigned networks. For example, negative links are available in signed networks and ignoring the negative links can result in over-estimation of the impact of positive links [7]; and most of triangles in signed networks satisfy balance theory [14]. However, these properties cannot be simply captured by unsigned network models. Hence, dedicated efforts are demanded to model signed networks. In this section, we propose a balanced signed Chung-Lu model (BSCL), which aims to preserve three important properties of signed networks – (1) degree distribution; (2) ratio of positive/negative links and (3) distribution of balanced and unbalanced triangles.

4.1 Balanced Signed Chung-Lu Model In this subsection, we first briefly introduce the basic Chung-Lu model and its variants, from which we build the proposed signed network model. The Chung-Lu (CL) model first takes as input an unsigned network $\mathcal{G}_I = (V_I, \mathcal{E}_I)$ and independently decides whether each of the N^2 edges are placed in the generated network with each edge e_{ij} having probability $\frac{d_i d_j}{2M}$. It can be shown that the expected degree distribution of the output network \mathcal{G}_o is equivalent to that of \mathcal{G}_I . A fast variant of the Chung-Lu model, FCL [12], proposed to create a vector π which consists of $2M$ values, where for each edge both incident vertices are added to the vector. Rather than selecting whether each of the N^2 edges get added to the network (as done in CL), FCL can just randomly sample two vertices from π , since this simulates the degree distribution. However, since most real-world unsigned networks have higher clustering coefficients than those generated by CL and FCL, another CL variant TCL was introduced in [11] to maintain the transitivity. Rather than always picking two vertices from π , instead, TCL occasionally picks a single vertex from π and then, based on a parameter ρ , performs a two-hop walk to select the second vertex. When including this edge, the process is explicitly constructing at least one triangle. Thus TCL aims to include more triangles into the network while maintaining the same expected distribution as the input

network.

We propose to construct our model based on the TCL model, which automatically allows the mechanism for maintaining the degree distribution and also the local clustering coefficient during the construction process. However, as previously mentioned, the distribution of formed triangles is a key property in signed networks and most of these triangles adhere to balance theory. Note that when performing the wedge closure procedure we are not only closing a single wedge, but there could be other common neighbors between these two vertices. Thus, we introduce a parameter β , which denotes the probability of assigning the edge sign to ensure the majority of closing wedges are balanced. With the introduction of this parameter, our model will be able to capture a range of balance in signed networks.

Meanwhile, we also want to maintain the distribution of signed links. However, the above process of determining the edge sign for wedge closure is based on balance theory (i.e., local sign perspective) and not on $\eta[+]$ (i.e., global sign perspective). Thus, this implies that when randomly inserting an edge into the network if we simply choose the sign based on $\eta[+]$, then this could lead to our generated networks deviating from sign distribution of the input networks. Therefore we introduce α which is a corrected sign distribution and is used to correct the bias of positive or negative edges that will be created through the use of β .

With the introduction of the two parameters, the proposed balanced signed Chung-Lu model (BSCL) is shown in Algorithm 1, where the first step is to construct \mathcal{E} , which will be used for calculating π , the sampling vector based on the degree distribution of \mathcal{G}_I when selecting which vertices to connect a random edge between. Next we calculate the properties of the input network we aim to preserve including the positive sign percentage $\eta[+]$, vector of degrees \mathbf{d} , and percentage of balanced triads Δ_B . Then we estimate the major parameters of BSCL including ρ , α and β . Finally we generate the network based on the learnt parameters. In the following subsections, we first discuss the network generation process, and then later discuss how these parameters can be automatically and efficiently learned.

4.2 Network Generation for BSCL Assuming we are given the set of parameters as shown in Algorithm 2, we generate a synthetic network maintaining the key signed network properties as follows. First, we use the FCL method for the construction of a set of edges \mathcal{E}_o , which adheres to the degree distribution found in the original network and contains M edges. This is used as a seed network, and we will one-by-one add a new edge to this set while removing the oldest, until all the original

edges from the FCL method have been removed. The reason is due to the fact when performing the wedge closure procedure, we need to have an initial set of edges to be closed into triangles that are balanced or not. Then we split the unsigned edges into two sets, \mathcal{E}_o^+ and \mathcal{E}_o^- , by randomly assigning edge signs such that the percentage of positive links matches that of the input network.

Algorithm 1 Balanced Signed Chung-Lu Model

Input: Signed Network $\mathcal{G}_I = (V_I, \mathcal{E}_I^+, \mathcal{E}_I^-)$

Output: Synthetic Signed Network $\mathcal{G}_o = (V_I, \mathcal{E}_o^+, \mathcal{E}_o^-)$

$\mathcal{E}_I = \mathcal{E}_I^+ \cup \mathcal{E}_I^-$

$\pi \leftarrow$ Sampling vector of degree distribution in \mathcal{E}_I

$\eta[+] \leftarrow \frac{M_I^+}{M_I^+ + M_I^-}$

$\mathbf{d} \leftarrow \text{Calculate_Degree_Vector}(V_I, \mathcal{E})$

$\Delta_B \leftarrow$ Percentage balanced triangles in \mathcal{G}_I

$\rho, \alpha, \beta \leftarrow \text{Parameter_Learning}(\mathcal{E}_I, \eta[+], \Delta_B, \mathbf{d}, M)$

$\mathcal{E}_o^+, \mathcal{E}_o^- \leftarrow \text{Network_Generation}(\eta[+], M, \pi, \rho, \alpha, \beta)$

Algorithm 2 Network_Generation($\eta[+], M, \pi, \rho, \alpha, \beta$)

$\mathcal{E}_o = \text{FCL}(M, \pi)$

$\mathcal{E}_o^+, \mathcal{E}_o^- \leftarrow$ Randomly partition \mathcal{E}_o based on $\eta[+]$

for 1 to M **do**

$v_i =$ sample from π

if *close_a_wedge*(ρ) **then**

$v_j =$ Perform two-hop walk from v_i

if *close_for_balance*(β) **then**

 Add e_{ij} to \mathcal{E}_o^+ or \mathcal{E}_o^- based on the sign that closes the wedge and common neighbors to have more balanced triangles

else

 Add e_{ij} to \mathcal{E}_o^+ or \mathcal{E}_o^- to have more unbalanced triangles

else insert a random edge

$v_j =$ sample from π

if *create_positive_edge*(α) **then**

$\mathcal{E}_o^+ \leftarrow \mathcal{E}_o^+ \cup \{e_{ij}\}$

else

$\mathcal{E}_o^- \leftarrow \mathcal{E}_o^- \cup \{e_{ij}\}$

 Remove oldest edge from $\{\mathcal{E}_o^+ \cup \mathcal{E}_o^-\}$ respectively

return $\mathcal{E}_o^+, \mathcal{E}_o^-$

Next, we add M new edges to the network, while removing the oldest edge in the network for each new edge inserted. This process makes use of the sampling vector, π , that is similarly used in FCL. The general idea is that we will first select a vertex v_i and next select v_j by either choosing to perform either a two-hop walk or randomly select another vertex from π . The probability of performing the two-hop walk is based on the parameter ρ , which is used to control the number of

triangles in the network.

If a two-hop walk is performed, then this explicitly created at least one triangle once the edge e_{ij} is inserted, but there is also the possibility that we are closing other triangles, since all common neighbors of v_i and v_j will also be closed. Thus, when determining the sign of the edge e_{ij} , we consider all these triangles. The parameter β is used to control the probability that when closing the triangle(s) if we select the edge sign that would adhere to balance theory, i.e., desiring more balanced than unbalanced relations amongst the common neighbors of v_i and v_j . In the opposite case, we would select the edge sign to disobey balance theory. We note this is necessary since not all signed networks are completely balanced, and in fact real-world networks can have a varied percentage of triangles balanced.

If not performing the explicit triangle closing step as described above, we insert a randomly selected edge based on the two vertices v_i and v_j selected from π . We next need to determine the edge sign for this new edge. This is performed by using α , the corrected sign distribution that is used to take into consideration the possible conflicting decisions for edge signs when selecting above for balance theory or not using the parameter β .

One step we did not mention in Algorithm 2 is that we also make use of the queue for when having collisions just as presented in FCL and TCL. The final step in the BSCL network generation process is to construct the output synthetic signed network $\mathcal{G}_o = (V_I, \mathcal{E}_o^+, \mathcal{E}_o^-)$.

4.3 Parameter Learning In the last subsections, we have introduced the BSCL model and network generation process based on the parameters ρ, β , and α , we now discuss how to learn these parameters from the input signed network. The motivation of the learning process is that without it, the model is actually not able to generate a synthetic network and would instead need the parameter values to always be hand-crafted for each network.

4.3.1 Learning ρ . For the fitting of the parameter ρ , we make use of the Expectation-Maximization learning method proposed in the TCL model. The general idea is that it can be learned after defining a hidden variable associated with each edge as to whether it was added to the network randomly or through a wedge being closed into a triangle. Details of this process can be found in [11].

4.3.2 Learning β . The solving of the balancing parameter β , involves first calculating the expected triangles that are created based on the wedge closing operation and also based on the random edge insertions. Based on the expected number of balanced triangles cre-

ated randomly, β will be picked such that the we can explicitly and correctly create more triangles to be balanced or not to achieve the overall proportion found in the input network. Since the starting set of edges are constructed with the FCL method, each edge will have been added into the network with probability $p_{ij} = \frac{d_i d_j}{2M}$, where d_i is the degree of v_i and M is the number of edges in the network. We note that the expected number of common neighbors between two vertices v_i and v_j would be equivalent to the number of triangles that get created if the edge e_{ij} were randomly inserted into the network; we denote this number of triangles to be Δ_{ij}^{random} .

To obtain the number of common neighbors, for each vertex $v_l \in V_I \setminus \{v_i, v_j\}$, we calculate the probability that v_l is a common neighbor of v_i and v_j based on the probability there exists an edge from v_l to both v_i and v_j . Note that after having the probability of the existence for the first edge e_{il} , we must subtract 1 from d_l , since we have already conditioned on the existence of the first edge e_{il} , thus causing v_l to have one less opportunity to connect to v_j . We formulate this idea as the following:

$$\begin{aligned}\Delta_{ij}^{random} &= \sum_{v_l \in V \setminus \{v_i, v_j\}} \left(\frac{d_i d_l}{2M} \right) \left(\frac{d_j (d_l - 1)}{2M} \right) \\ &= \left(\frac{d_i d_j}{2M} \right) \sum_{v_l \in V \setminus \{v_i, v_j\}} \left(\frac{d_l (d_l - 1)}{2M} \right)\end{aligned}$$

Next we present the average value of Δ_{ij}^{random} across all possible unordered pairs of vertices as follows:

$$\Delta_{\mathcal{G}}^{random} = \frac{1}{\frac{1}{2}N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N \Delta_{ij}^{random}$$

where $\Delta_{\mathcal{G}}^{random}$ is used to denote the average triangles constructed by an inserted edge in the CL model. We note that the above would require $O(N^3)$ time to compute, but using the fact that $2M = N * avg(d)$, we can use the following approximation if we treat the summation such that it includes v_i and v_j instead of excluding them. We use $avg(d)$ to denotes the average degree and $avg(d^2)$ represents the average value of squared degrees.

$$\begin{aligned}\sum_{v_l \in V \setminus \{v_i, v_j\}} \left(\frac{d_l (d_l - 1)}{2M} \right) &\approx \sum_{v_l \in V} \left(\frac{d_l (d_l - 1)}{2M} \right) \\ &= \frac{(d_1^2 + d_2^2 + \dots + d_N^2) - (d_1 + \dots + d_N)}{2M} \\ &= \frac{(N * avg(d^2)) - (N * avg(d))}{N * avg(d)} \\ (4.1) \quad &= \left(\frac{avg(d^2) - avg(d)}{avg(d)} \right)\end{aligned}$$

We therefore rewrite $\Delta_{\mathcal{G}}^{random}$ as follows:

$$(4.2) \quad \Delta_{\mathcal{G}}^{random} \approx \frac{avg(d^2) - avg(d)}{avg(d)MN(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N d_i d_j$$

Since we only need to compute the averages once (which take $O(N)$) and then iterate over all pairs performing a constant time multiplication, we are able to calculate $\Delta_{\mathcal{G}}^{random}$ much faster with the approximation.

If we further analyze the wedges (i.e., common neighbors) to be one of the following formations: $(\{++\}, \{+-\}, \{-+\}, \{--\})$, where $\{+-\}$ is used to represent the wedge formed by edges e_{il} and e_{lj} where their signs are positive and negative, respectively. As we will attempt to correctly maintain this sign distribution with the parameter α , we can assume that all wedges were created by the original sign distribution. Below, we use $\Delta_{ij}^{random+-}$ to represent the number of wedges that would be closed into triangles when adding the edge e_{ij} and were formed with a wedge of type $\{+-\}$; similar definitions for the other wedge types are:

$$\begin{aligned}\Delta_{ij}^{random++} &= \Delta_{ij}^{random} \times \eta[+]\eta[+] \\ \Delta_{ij}^{random+-} &= \Delta_{ij}^{random-+} = \Delta_{ij}^{random} \times \eta[+](1 - \eta[+]) \\ \Delta_{ij}^{random--} &= \Delta_{ij}^{random} \times (1 - \eta[+])(1 - \eta[+])\end{aligned}$$

The expected number of balanced triangles that would be created if the edge e_{ij} is created can be obtained via the expected number of wedges of different types and the corrected positive sign probability α as:

$$(4.3) \quad \begin{aligned}\Delta_{ijB}^{random} &= \alpha \Delta_{ij}^{random++} + (1 - \alpha) \Delta_{ij}^{random+-} \\ &\quad + (1 - \alpha) \Delta_{ij}^{random-+} + \alpha \Delta_{ij}^{random--}\end{aligned}$$

where for a wedge with two existing edges to close to a balanced triangle, there would need to be an even number of negative links according to balance theory. This can also be extended for the calculation of $\Delta_{\mathcal{G}B}^{random}$.

Similarly we will calculate the expected total number of triangles and the balanced percentage when using the wedge closure edge insertion method (as compared to the random edge insertion). The main idea for this wedge closure is that we are guaranteed to select vertices such that we have at least one triangle being created each time. We then need to also add the expected number of triangles that would be created just randomly based on the degree (similar to the random edge insertion case above). Note that we must discount the degree of v_i and v_j by 1, since in this method we have already explicitly used one of the links coming from both to discover this one common neighbor for the wedge closure edge insertion. Let us denote the selected common neighbor as v_c , which forms a wedge with the edges e_{ic} and e_{cj} .

$$\begin{aligned}\Delta_{ij}^{triangle} &= 1 + \sum_{v_l \in V \setminus \{v_i, v_j, v_c\}} \left(\frac{(d_i - 1)d_l}{2M} \right) \left(\frac{(d_j - 1)(d_l - 1)}{2M} \right) \\ &= 1 + \left(\frac{(d_i - 1)(d_j - 1)}{2M} \right) \sum_{v_l \in V \setminus \{v_i, v_j, v_c\}} \left(\frac{d_l (d_l - 1)}{2M} \right)\end{aligned}$$

We can make a similar assumption as Eq. (4.1), and can simplify the formulation of $\Delta_{ij}^{triangle}$ as:

$$\Delta_{ij}^{triangle} \approx 1 + \left(\frac{d_i d_j}{2M} - \frac{d_i + d_j}{2M} \right) \left(\frac{avg(d^2) - avg(d)}{avg(d)} \right)$$

Then we can also calculate $\Delta_{\mathcal{G}}^{triangle}$ similar to $\Delta_{\mathcal{G}}^{random}$. The parameter β decides whether or not to close the triangle to be balanced or not, thus, the expected number of balanced triads is:

$$(4.4) \quad \Delta_{\mathcal{G}_B}^{triangle} = \beta \Delta_{\mathcal{G}}^{triangle}$$

If we calculate the balance of the input network, Δ_B , we can then calculate the optimal value of β based on the original balance. Therefore using Eqs. (4.3) and (4.4), we have the following:

$$\Delta_B = \frac{\Delta_{\mathcal{G}_B}^{triangle} + \Delta_{\mathcal{G}_B}^{random}}{\Delta_{\mathcal{G}}^{random} + \Delta_{\mathcal{G}}^{random}}$$

and therefore, β can be estimated as:

$$\beta = \frac{\Delta_B(\Delta_{\mathcal{G}}^{triangle} + \Delta_{\mathcal{G}}^{random}) - \Delta_{\mathcal{G}_B}^{random}}{\Delta_{\mathcal{G}}^{triangle}}$$

4.3.3 Learning α . The parameter α is used for determining a corrected sign distribution due to the disruption caused when selecting the edge sign based on balance theory with the parameter β . Note that we can't naively use the percentage of positive links from the original network due to the fact that the wedge closure process is selecting signs that aren't based on the sign distribution and instead based locally towards making them balanced or not. Therefore we must learn the corrected percentage to be used when taking into account the expected percentage of triangle closures that result in positive links. We directly calculate the probability for each of the wedge types based on $\eta[+]$, since the initial network based on FCL would have had an exact sign distribution equal to that of the input networks, and we seek to maintain this throughout the BSCL process. We then multiply by either β or $(1 - \beta)$ based on the probability that would have resulted in the inserted wedge closing edge to be positive and the percentage of edges in BSCL that are positive can be estimated as:

$$\begin{aligned} & \rho \left(\beta \left((\eta[+] * \eta[+]) + (1 - \eta[+]) * (1 - \eta[+]) \right) \right. \\ & \quad \left. + (1 - \beta) \left((\eta[+] * (1 - \eta[+])) + (\eta[+] * (1 - \eta[+])) \right) \right) \\ & \quad + (1 - \rho) \alpha \end{aligned}$$

where the first and second terms are representing the closing of triangles and thus first have a probability of ρ . Specifically the first term is closing the wedges $\{+, +\}$ and $\{-, -\}$ to be balanced and thus multiplied by β and the second term is closing the wedges $\{+, -\}$ and $\{-, +\}$ with a positive, but to be unbalanced and hence multiplied by $(1 - \beta)$. The last term is representing the random insertion of a positive edge using the parameter α that we seek to solve for.

We next set the expected percentage of edges in BSCL that are positive equal to $\eta[+]$, which is the desired percentage we desire the output network to maintain. We can then obtain α as:

$$\begin{aligned} \alpha = \frac{1}{(1 - \rho)} & \left(\eta[+] - \left(\rho \beta \left((\eta[+] * \eta[+]) + ((1 - \eta[+]) * (1 - \eta[+])) \right) \right. \right. \\ & \quad \left. \left. + (1 - \beta) \left((\eta[+] * (\eta[-])) + (\eta[+] * (\eta[-])) \right) \right) \right) \end{aligned}$$

As mentioned before, the parameters α and β are not independent of each other since they can both influence the distribution of formed triangles and signed links. Thus, we will perform an alternative updating scheme until convergence for these two parameters β and α .

4.4 Time Complexity We first discuss the running time of the learning algorithm and then the time needed for the network generation process. The preprocessing needed for the learning algorithm is to determine $\eta[+]$ and Δ_B of the original network. We can determine $\eta[+]$ trivially in $O(M)$. However, Δ_B can be reduced the the complexity of triangle listing algorithms, which can be easily performed using classical methods in $O(\min(M^{3/2}, Md_{max}))$, where d_{max} is the maximum degree in the network[13]. The learning process for the parameter ρ has been shown in [11] to be $O(sI)$ where I is the number of iterations in the EM method and s is the number of edges sampled for each iteration. The running time of β and α can be determined as follows. We initially calculate the expected number of triangles added by each of the processes of BSCL, which takes $O(N^2)$. With sampling techniques this could be reduced, but we leave this as one future work. The update to both β and α are then calculated in $O(1)$. Thus when allowing for I' maximum iterations of the alternating update process we have the overall learning for BSCL is $O(\min(m^{3/2}, md_{max}) + N^2 + sI + I')$. The generation process of BSCL, is built upon the fact that the running time for TCL is shown to be $O(N + M)$ in [11]. The triangle closing process of determining the best sign selection based on the set of triangles being closed, this is reduced to the complexity of common neighbors between two vertices, which is known to be $O(d_{max}^2)$. When averaging over the M edge insertions, this can be represented as $O(Md_{avg}^2)$. Thus the generation process of BSCL is $O(N + Md_{avg}^2)$.

5 Experiments

In this section, we conduct experiments to evaluate the effectiveness of the proposed signed network model. In particular, we try to answer two questions via our experiments - (1) can the proposed model, BSCL, effectively maintain signed network properties? and (2) is the parameter learning algorithm able to learn the appropriate parameter values from the input signed network? We first introduce the datasets used, then design experiments to seek answers for the above two questions by first comparing BSCL against baseline models, and finally we perform an analysis of the parameter learning algorithm of BSCL.

Table 1: Statistics of three signed social networks.

Network	N	(E, $\eta[+]$)	Δ_B
Bitcoin-Alpha	3,784	(14,145 , 0.915)	0.862
Bitcoin-OTC	5,901	(21,522 , 0.867)	0.869
Epinions	131,580	(711,210 , 0.830)	0.892

5.1 Datasets For our study of signed network modeling, we collect three signed network datasets, i.e., Bitcoin-Alpha, Bitcoin-OTC, and Epinions. We provide more details of the datasets in Table 1.

The Bitcoin-Alpha and Bitcoin-OTC datasets were collected from Bitcoin Alpha¹ and Bitcoin OTC², respectively. In these sites, users can buy and sell things in an open marketplace using Bitcoins. For the safety of the users, they form online trust networks to protect against scammers. Although these are weighted signed networks, we converted all the positive and negative links to 1s and -1s, respectively. We also have collected a dataset from Epinions³, which is a product review site where users can mark their trust or distrust to other users, representing positive and negative links in the signed network, respectively. Note that, since we focus on undirected signed networks, we ignore the directions of signed links in these four datasets.

5.2 Network Generation Experiments The first set of experiments are to compare the network properties of the resulting generated networks from our model and the baselines. These properties will be used as a metric to determine how well each of the models is able to capture the underlying dynamics of signed networks. More specifically, we will focus on the three key signed network properties - (1) degree distribution; (2) positive/negative link ratio and (3) proportion of balance/unbalanced triangles suggested by balance theory. Note that we also present the local clustering coefficient distribution and the triangle distribution. Our results are the averaged results of 10 generated networks for each of the methods on each dataset.

The first group of two baselines are existing signed network models, but not generative: (1) **Ants14**: it is an interaction-based model for signed networks based on using ants to lay pheromone on edges and it has local and global processes that make balanced triangles and also create random longer range connections [17]; and (2) **Evo07**: it is an evolutionary model for signed networks that had a “friendliness” index that controls the probability of positive or negative links and also a parameter that controls the maximum amount of

unbalance that any particular node could have in the network [8]. Note that for Ants14, we perform a grid search on the parameter space for its 6 parameters according to the values reported in their paper[17]. Similarly, for Evo07, a grid search was performed for the two parameters and we report the best solution for each dataset.

The next three baselines are built upon two popular unsigned generative models. We first convert the network to unsigned by ignoring the links, run the baseline model, and then randomly assign signs to the edges such that the global sign distribution is maintained using $\eta[+]$. They are - (1) **SFCL** from FCL [11]; (2) **STCL** from the TCL model [11]; and (3) **SKron** from the Kronecker Product model [5].

For our model, BSCL, we present the results of the following two variants: (1) **BSCL**: it is our balanced signed Chung-Lu model where the parameters are learned from the given learning algorithm; and (2) **BSCLf**: it is the “fixed” parameter model where the parameters of α and β are not learned, but rather heuristically set equal to $\eta[+]$ and Δ_B , respectively.

The results of the properties that are in common with unsigned networks (i.e., the degree distribution and the local clustering coefficient) can be seen in Figure 1. We see that BSCL, BSCLf, STCL, and SFCL all perform near identically on the degree distribution as they are all Chung-Lu based models and therefore can correctly maintain the degree distribution. However, it can be seen that the two signed network baselines, Ants14 and Evo07, perform very poorly and do not even appear to follow a power-law distribution. We mention that SKron is not able to exactly model the degree distribution, but does not perform as poorly as the two existing signed network baselines. For the existing signed network models similar poor findings can be found for the local clustering coefficient. Our proposed model along with STCL perform the best, since they share the same ρ parameter. The SKron and SFCL models have some clustering, but not near enough to be close to the original input network.

In Table 2, we can show the positive/negative link ratio; while in Table 3, we present the proportion balance/unbalance triangles. The results in Table 4 provide a fine-grained comparison by separating the four types of triads on Bitcoin OTC dataset. In Bitcoin Alpha dataset, our model BSCL is able to achieve the closest proportion of balance triangles. Then in the Bitcoin OTC dataset, the Ants14 performs the best in terms of the proportion of balance in the network, but we note that in Table 4, they only achieve this by drastically changing the distribution among the four triangle types, thus leaving our model to perform

¹<http://www.btcalpha.com>

²<http://www.bitcoin-otc.com>

³<http://www.epinions.com>

Table 2: Positive/Negative Link Sign Distribution.

Links Positive	Real	Ants14	Evo07	SFCL	STCL	SKron	BSCL	BSCLf
Bitcoin Alpha	0.915	0.741	0.917	0.915	0.915	0.913	0.912	0.879
Bitcoin OTC	0.867	0.740	0.869	0.867	0.867	0.865	0.860	0.821
Epinions	0.830	0.940	0.830	0.830	0.830	0.832	0.808	0.753

Table 3: Proportion of Triangles Balanced.

Percent Balanced	Real	Ants14	Evo07	SFCL	STCL	SKron	BSCL	BSCLf
Bitcoin Alpha	0.840	0.787	0.788	0.786	0.784	0.776	0.802	0.754
Bitcoin OTC	0.858	0.815	0.688	0.697	0.698	0.687	0.747	0.698
Epinions	0.892	0.939	0.693	0.645	0.644	0.643	0.747	0.681

Table 4: Distribution of Triangle Types in Bitcoin OTC.

Triad Type	Real	Ants14	Evo07	SFCL	STCL	SKron	BSCL	BSCLf
+++	0.720	0.448	0.625	0.651	0.652	0.637	0.701	0.627
++-	0.133	0.172	0.312	0.300	0.299	0.310	0.252	0.299
+--	0.138	0.367	0.062	0.046	0.046	0.050	0.046	0.072
---	0.009	0.013	0.000	0.002	0.002	0.003	0.002	0.003

the best overall in terms of the triangle distribution. Note that in both of the Bitcoin datasets, our model BSCL is able to achieve better performance than the baselines in terms of the triangle distributions, but only at the expense of sacrificing $< 1\%$ in terms of the positive/negative link ratio away from the true link sign distribution. The results are similar in the Epinions dataset.

Thus we have partially answered the first question, mainly we have shown that BSCL is able to correctly model the Bitcoin datasets and still has the best performance on the Epinions dataset.

5.3 Parameter Learning Experiments The second set of experiments are designed to test the learning algorithm we have proposed in determining appropriate parameters for BSCL. Specifically, we perform a grid search across a reasonable area of the parameter space for α and β to obtain optimal parameters. Then we compare the performance of the learnt parameters and the searched optimal parameters to demonstrate the ability of the proposed parameter learning algorithm. For these experiments, we leave ρ unchanged (i.e., using its EM learning method), since we only want to test our introduced parameters.

We only present the results in Figures 2 in terms of percentage of balanced triangles and positive/negative link ratio for the Bitcoin Alpha dataset, since we have similar observations for Bitcoin OTC and Epinions with other settings. Note that the z-axis is the absolute difference away from the true input networks value (where lower is better). The “stars” in the figures are the coordinates along the x- and y-axis for the learned parameter. From the figures, it looks convincing that indeed our parameter learning algorithm is able to find

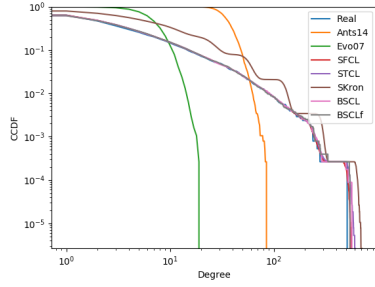
appropriate parameters for the input network.

6 Conclusion

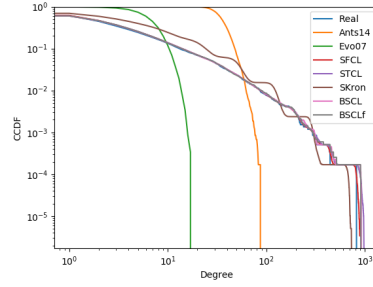
Recently there has been a growing interest in signed networks and in parallel the demand of fundamentally understanding large real-world networks has grown due to how ubiquitous they have become in today’s world. However, very few works have focused modeling of signed networks, as most have focused on unsigned networks. Generative network models specifically have been shown to provide a vast amount of insight into the underlying network structures. In signed networks specifically, social theories have been developed to describe the dynamics and mechanisms that drive the structural appearance of signed networks. Structural balance theory is one such social theory. We have proposed our Balanced Signed Chung-Lu model (BSCL) with the objective of preserving three key properties of signed networks - (1) degree distribution; (2) positive/negative link ratio and (3) proportion of balance/unbalanced triangles suggested by balance theory. To achieve this we introduced a triangle balancing parameter and a sign balancing parameter to control the distribution of formed triangles and signed links, respectively. A automated estimation approach for the two parameters allows BSCL to take as input a signed network, learn appropriate parameters needed to model the key properties, and then output a similar network maintaining the desired properties. We will further investigate both directed and weighted signed networks in future work.

Acknowledgements

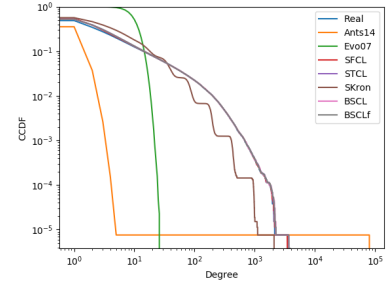
This material is based upon work supported by, or in part by, the National Science Foundation (NSF) under grant number IIS-1714741 and IIS-1715940.



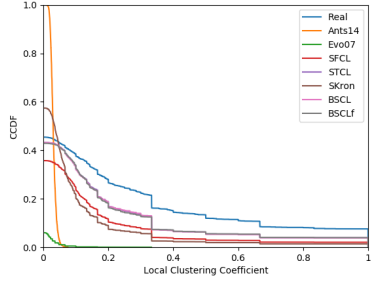
(a) Bitcoin Alpha Degree Dist.



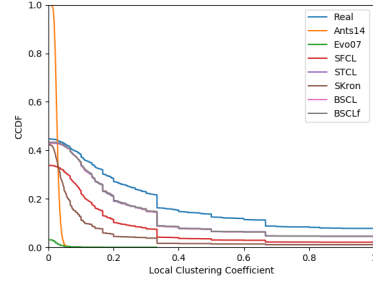
(b) Bitcoin OTC Degree Dist.



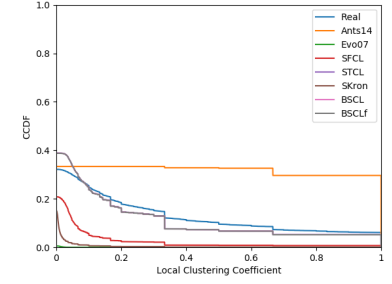
(c) Epinions Degree Dist.



(d) Bitcoin Alpha Local Clustering

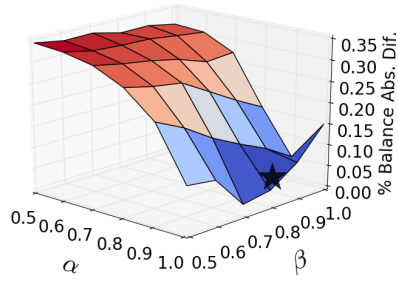


(e) Bitcoin OTC Local Clustering

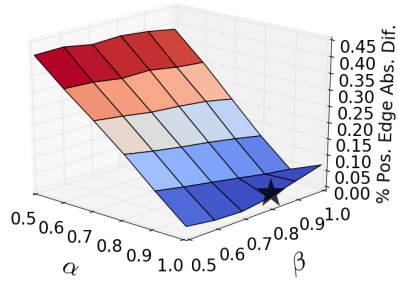


(f) Epinions Local Clustering

Figure 1: Degree Distribution and Local Clustering Coefficient.



(a) Bitcoin Alpha
% Balanced Triangles



(b) Bitcoin Alpha
Positive/Negative Ratio

Figure 2: Parameter Learning Analysis.

References

- [1] A.-L. BARABÁSI AND R. ALBERT, *Emergence of scaling in random networks*, science, 286 (1999), pp. 509–512.
- [2] D. CARTWRIGHT AND F. HARARY, *Structural balance: a generalization of heider's theory.*, Psychological review, 63 (1956), p. 277.
- [3] F. CHUNG AND L. LU, *The average distances in random graphs with given expected degrees*, Proceedings of the National Academy of Sciences, 99 (2002), pp. 15879–15882.
- [4] F. HEIDER, *Attitudes and cognitive organization*, The Journal of psychology, 21 (1946), pp. 107–112.
- [5] J. LESKOVEC, D. CHAKRABARTI, J. KLEINBERG, C. FALOUTSOS, AND Z. GHAHRAMANI, *Kronecker graphs: An approach to modeling networks*, Journal of Machine Learning Research, 11 (2010), pp. 985–1042.
- [6] J. LESKOVEC, J. KLEINBERG, AND C. FALOUTSOS, *Graphs over time: densification laws, shrinking diameters and possible explanations*, in Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, ACM, 2005, pp. 177–187.
- [7] Y. LI, W. CHEN, Y. WANG, AND Z.-L. ZHANG, *Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships*, in Proceedings of the sixth ACM international conference on Web search and data mining, ACM, 2013, pp. 657–666.

- [8] M. LUDWIG AND P. ABELL, *An evolutionary model of social networks.*, European Physical Journal B-Condensed Matter, 58 (2007).
- [9] M. MALEKZADEH, M. FAZLI, P. J. KHALILABADI, H. RABIEE, AND M. SAFARI, *Social balance and signed network formation games*, 2011.
- [10] S. MUSSMANN, J. MOORE, J. J. PFEIFFER, III, AND J. NEVILLE, *Incorporating assortativity and degree dependence into scalable network models*, in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, AAAI Press, 2015, pp. 238–246.
- [11] J. J. PFEIFFER, T. LA FOND, S. MORENO, AND J. NEVILLE, *Fast generation of large scale social networks while incorporating transitive closures*, in Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom), IEEE, 2012, pp. 154–165.
- [12] A. PINAR, C. SESHADHRI, AND T. G. KOLDA, *The similarity between stochastic kronecker and chung-lu graph models*, in Proceedings of the 2012 SIAM International Conference on Data Mining, SIAM, 2012, pp. 1071–1082.
- [13] T. SCHANK AND D. WAGNER, *Finding, counting and listing all triangles in large graphs, an experimental study.*, in WEA, Springer, 2005.
- [14] M. SZELL, R. LAMBIOTTE, AND S. THURNER, *Multirelational organization of large-scale social networks in an online world*, Proceedings of the National Academy of Sciences, 107 (2010), pp. 13636–13641.
- [15] J. TANG, X. HU, AND H. LIU, *Is distrust the negation of trust?: the value of distrust in social media*, in Proceedings of the 25th ACM conference on Hypertext and social media, ACM, 2014, pp. 148–157.
- [16] E. VOLZ AND L. A. MEYERS, *Susceptible–infected–recovered epidemics in dynamic contact networks*, Proceedings of the Royal Society of London B: Biological Sciences, 274 (2007), pp. 2925–2934.
- [17] V. VUKAŠINOVIĆ, J. ŠILC, AND R. ŠKREKOVSKI, *Modeling acquaintance networks based on balance theory*, International Journal of Applied Mathematics and Computer Science, 24 (2014), pp. 683–696.