2018 IEEE International Conference on Data Mining

# Signed Graph Convolutional Networks

Tyler Derr
*Data Science and Engineering Lab*
*Michigan State University*
derrtyle@msu.edu

Yao Ma
*Data Science and Engineering Lab*
*Michigan State University*
mayao4@msu.edu

Jiliang Tang
*Data Science and Engineering Lab*
*Michigan State University*
tangjili@msu.edu

***Abstract*—Due to the fact much of today's data can be represented as graphs, there has been a demand for generalizing neural network models for graph data. One recent direction that has shown fruitful results, and therefore growing interest, is the usage of graph convolutional neural networks (GCNs). They have been shown to provide a significant improvement on a wide range of tasks in network analysis, one of which being node representation learning. The task of learning low-dimensional node representations has shown to increase performance on a plethora of other tasks from link prediction and node classification, to community detection and visualization. Simultaneously, signed networks (or graphs having both positive and negative links) have become ubiquitous with the growing popularity of social media. However, since previous GCN models have primarily focused on unsigned networks (or graphs consisting of only positive links), it is unclear how they could be applied to signed networks due to the challenges presented by negative links. The primary challenges are based on negative links having not only a different semantic meaning as compared to positive links, but their principles are inherently different and they form complex relations with positive links. Therefore we propose a dedicated and principled effort that utilizes balance theory to correctly aggregate and propagate the information across layers of a signed GCN model. We perform empirical experiments comparing our proposed signed GCN against state-of-the-art baselines for learning node representations in signed networks. More specifically, our experiments are performed on four real-world datasets for the classical link sign prediction problem that is commonly used as the benchmark for signed network embeddings algorithms.**

***Index Terms*—Signed Networks, Graph Convolutional Networks, Network Embedding, Balance Theory**

## I. Introduction

Recently there has been a large and growing interest of generalizing neural network models to structured data, with one of the most prevalent structures being graphs (such as those found in social media). The idea of generalizing neural network models to graph structures has lately started to become more developed by overcoming the difficulties and trade-offs previously associated with fast heuristics compared to slow and more principled approaches. Graph convolutional neural networks (GCNs) [1]–[7] are modeled after the classical convolutional neural networks [8]. The first GCN introduced for learning representations at the node level was in [4], where they utilized GCNs for the semi-supervised node classification problem. Furthermore, learning low-dimensional node representations have been previously proven to be useful in many network analysis tasks beyond node classification [9], such as link prediction [10], community detection [11], and visualization [12].

Previous work has mostly focused on using GCNs for unsigned graphs (or graphs consisting of only positive links). However, especially with the ever growing popularity of online social media, signed graphs are becoming increasingly ubiquitous. This naturally leads the question as to whether unsigned GCNs are suitable to be used on signed networks. Unfortunately, there are many reasons as to why unsigned GCNs are not capable of learning meaningful node representations in signed networks. First, it is unclear how they would handle the availability of negative links in signed networks, and furthermore, negative links invalidate some of the underlying key assumptions of GCNs. For example, GCNs designed for unsigned networks learn a node representation using the fundamental social theory homophily [13], which states users having connections are more likely to be similar than those without links. Hence, the aggregation processes of GCNs use local neighborhood information when constructing the low-dimensional embedding for each node. However, homophily may not be applicable to signed networks [14]. Instead, in signed networks, there are specific social theories and principles defined in the context of having both positive and negative links. Therefore dedicated efforts are needed for redesigning GCNs specifically for signed networks.

Although it is now clear that GCNs will need to be specifically redesigned to provide the same fruitful performance as previously shown in unsigned networks when applied to signed networks, there are still tremendous challenges to overcome. When designing signed GCNs the primary challenges are: (1) how to correctly handle negative links, since their properties are inherently different than those of positive links; and (2) how to combine the positive and negative links into a single coherent model to learn effective node representations. Thus, we turn our attention towards social theories specific to signed networks (similarly to how the unsigned models were constructed using unsigned theories like homophily). More specifically, one fundamental signed network social theory that had been developed in social psychology is balance theory [15], [16]. If we can harness the power of this signed network social theory, which provides a better understanding of negative links and how they form complex relations with positive links, then this offers the opportunity to solve these two challenges when incorporating these ideas into our framework of designing a signed GCN. Our major contributions are listed as follows:

978-1-5386-9159-5/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDM.2018.00113

- Propose a Signed Graph Convolutional Network (SGCN) that is constructed based on balance theory to correctly integrate negative links during the aggregation process;
- Construct an objective function for our SGCN based on signed network social theories to easily learn an effective low-dimensional representation for each node in the network; and
- Conduct experiments on four real-world signed networks to comprehensively demonstrate the effectiveness of our proposed SGCN framework.

## II. The Proposed Framework

Graph convolutional neural networks have recently started to become more developed and have already shown their superiority in extracting and aggregating information from graph data. Their use cases spread over the vast field of network analysis, but one such domain that has shown to be very influential recently is network embedding. The discovery of representative low-dimensional features for each node in the network has previously shown to enhance many tasks from link prediction and node classification, to community detection and visualization. However, previous work has mostly focused on constructing GCNs for unsigned networks. Due to the inherent differences between unsigned and signed networks, this leaves a gap that we seek to bridge with the development of a signed graph convolutional network (SGCN).

Even with dedicated efforts towards the construction of a GCN specific to signed networks, there are still tremendous challenges we must face and overcome. The first of which is figuring out how we can correctly incorporate negative links during the aggregation process. We cannot simply treat the negative links the same as positive links, since their properties and semantic meaning vastly differ. The second challenge is how we can combine the two sets of links (i.e., positive and negative) into a single coherent model. This combination is essential because certainly positive and negative links interact in the network structure in complex ways and indeed are not segregated and isolated from each other.

In this work we propose to go to the roots of signed network analysis and utilize one of the most fundamental and indispensable signed social theories developed in social psychology, balance theory [15], [16]. We harness balance theory to construct a bridge to connect the gap between the ongoing development of GCNs for unsigned networks and signed networks. In the remainder of this section we will first briefly discuss a general GCN framework in the unsigned network setting and discuss the relationships of this framework to the structure of signed networks. Then we introduce balance theory and how we can use this signed social theory to correctly capture both positive and negative links simultaneously during the aggregation process. Thereafter, we present how to learn the parameters of our SGCN – first through the construction of an objective function designed to effectively learn the node representations in signed networks, and finally discussing the optimization procedure taken to optimize our proposed objective.

Before introducing the details of the framework, we introduce some definitions and notations that will be used throughout this paper. Let $\mathcal{G} = (\mathcal{U}, \mathcal{E}^+, \mathcal{E}^-)$ be a signed network, where $\mathcal{U} = \{u_1, u_2, \dots u_n\}$ represents the set of $n$ nodes while $\mathcal{E}^+ \subset \mathcal{U} \times \mathcal{U}$ and $\mathcal{E}^- \subset \mathcal{U} \times \mathcal{U}$ denote the sets of positive and negative links, respectively, where $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$. We use $\mathbf{A} \in \mathbb{R}^{n \times n}$ to denote the adjacency matrix of the signed network $\mathcal{G}$, where $\mathbf{A}_{ij} = 1$ means there exists a positive link from $u_i$ to $u_j$, $\mathbf{A}_{ij} = -1$ denotes a negative link, and $\mathbf{A}_{ij} = 0$ otherwise (meaning no link from $u_i$ to $u_j$).

### A. Unsigned Graph Convolutional Networks

Currently, most GCNs have a similar structure in that they utilize a convolutional operator that can share weights across all locations in the graph. The benefits of this neural network structure in graphs as compared to the cumbersome fully connected models are at least three fold: 1) it avoids the parameter explosion associated with fully connected layers; 2) it allows for parameter sharing across the network to avoid overfitting; and 3) a single GCN is capable of handling as input graphs of varying structures and even sizes. Note that the aggregation functions are the components for GCNs where they primarily differ.

**Limitations of unsigned GCN for signed networks:** Given the above discussion on the unsigned GCN framework, we note that in relation to signed networks this would be similar to applying the unsigned GCN on the positive only adjacency matrix $\mathbf{A}^+$ where $\mathbf{A}^+_{ij} = 1$ if there exists a positive link between users $u_i$ and $u_j$, and 0 otherwise. However, this would ignore the negative links.

Initially, our thoughts may lead to some naïve approaches of handling the negative links by either ignoring them, treating them the same or the negation of positive links, or separately applying the GCN framework to first the positive network, and then the negative network with finally combining them at the end stage. However, each of these methods are either based on incorrect assumptions or ignoring parts of the rich information awaiting to be extracted from the complex network structure of signed networks. For example, trivially treating the negative links the same as the positive links would be an incorrect assumption, since negative links have been shown to have different principles and semantically represent vastly different meanings. Similarly, treating negative links as the negation of positive links is likely an incorrect assumption [14]. This leaves the last two initial thoughts of ignoring the negative links or applying an unsigned GCN separately on the positive only and negative only networks, but intuitively this ignores a large amount of information including signed social theories [15]–[17]. Therefore, next we will discuss one such signed social theory, balance theory [15], [16] and how we propose to harness it for capturing both the positive and negative links coherently together during the aggregation process.

### B. Aggregation paths with positive and negative links

Balance theory [15], [16] implies "the friend of my friend is my friend" and "the foe of my friend is my foe". The theory classifies cycles in a signed network as being either

*balanced* or *unbalanced*, where a balanced (or unbalanced) cycle consist of an even (or odd) number of negative links. Similarly, we denote a *balanced path* (or *unbalanced path*) as one that consists of an even (or odd) number of negative links. Therefore we can see that if a path of length $l$ from $u_i$ to $u_j$ has an even number of negative links, then balance theory would suggest a positive link between $u_i$ an $u_j$. We place the users that reach $u_i$ along a balanced (or unbalanced) path of length $l$ in the set $B_i(l)$ (or $U_i(l)$). In Figure 1 we provide an illustration of how all the signed paths of a given length would place users along paths from $u_i$ into their respective sets.

Before continuing, we define $\mathcal{N}_i^+$ to be the set of positive neighbors of a user $u_i$, i.e., $u_j \in \mathcal{N}_i^+$ if $\mathbf{A}_{ij} = 1$. We similarly denote the set of negative neighbors for user $u_i$ as $\mathcal{N}_i^-$, where $u_j \in \mathcal{N}_i^-$ when $\mathbf{A}_{ij} = -1$. In Figure 1 we can see that when having a balanced path of length $l$ from $u_i$ to some user $u_k$ (i.e., $u_k \in B_i(l)$), then all the positively linked neighbors of $u_k$ (which we denoted as the set $\mathcal{N}_k^+$) would be placed in $B_i(l+1)$. This is because adding a positive link to a balanced path still results in a balanced path Similarly when adding a negative link to a balanced path, we obtain an unbalanced path.

From Figure 1 we can also obtain the balanced and unbalanced sets $B_i(l+1)$ and $U_i(l+1)$ of length $l+1$, respectively for user $u_i$, from the sets $B_i(l)$ and $U_i(l)$ of length $l$. A recursive definition for calculating the balanced and unbalanced sets from the perspective of user $u_i$ as follows:

$$\text{When } l = 1, \quad B_i(1) = \{u_j \mid u_j \in \mathcal{N}_i^+\}; \quad U_i(1) = \{u_j \mid u_j \in \mathcal{N}_i^-\}.$$

$$\begin{aligned} \text{For } l > 1, \quad B_i(l+1) = &\{u_j \mid u_k \in B_i(l) \text{ and } u_j \in \mathcal{N}_k^+\} \\ &\cup \{u_j \mid u_k \in U_i(l) \text{ and } u_j \in \mathcal{N}_k^-\} \\ U_i(l+1) = &\{u_j \mid u_k \in U_i(l) \text{ and } u_j \in \mathcal{N}_k^+\} \\ &\cup \{u_j \mid u_k \in B_i(l) \text{ and } u_j \in \mathcal{N}_k^-\}. \end{aligned} \tag{1}$$

We note that these definitions, based upon balance theory, now allow us a principled way of aggregating and propagating information in signed networks using balanced and unbalanced paths/sets. Next we will propose aggregation functions for our signed GCN and follow with the rest of the details of our framework.

### C. Signed Graph Convolutional Network

Before formalizing our signed graph convolutional network, we provide some insights and intuitions behind the construction. First, in unsigned GCNs, when constructing a node representation, they aggregate their immediate local neighbors' information into a single representation and then through the use of multiple layers, propagate this in the network allowing a node to incorporate information from a multi-hop neighborhood . However, in signed networks, not all neighbors are the same. This is because semantically users that are connected through positive (or negative) links are "friends" (or "foes"). Similarly, for users in $u_i$'s balanced (or unbalanced) sets, balance theory would suggest they are their "friends" (or "foes"). Therefore, we propose rather than maintaining a single representation for each node, we keep a representation

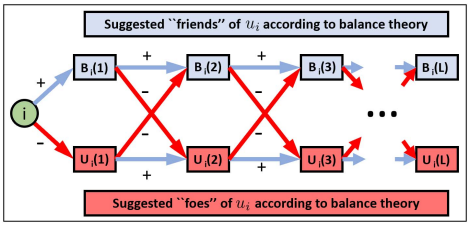


Fig. 1. An illustration of the link structure that leads to users being used in the different aggregations based on balance and unbalanced paths.

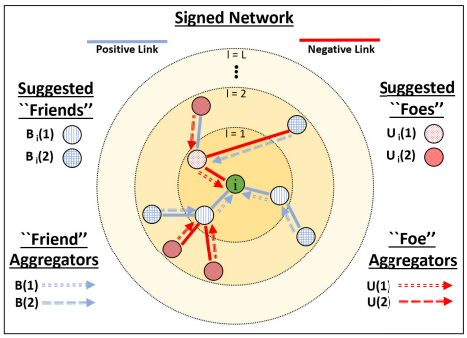


Fig. 2. Illustrating how SGCN aggregates neighbor information.

of both their "friends" and "foes", which provides a more thorough representation of a given user.

In Figure 2 we provide an illustration of how we plan to aggregate and propagate information in signed networks. Note that the circles labeled $l = 1, 2, \ldots, L$ are used to denote how many hops away the user is from $u_i$ and simultaneously denotes at which layer in our signed GCN that user's information will be incorporated. We can observe that we could have a separate aggregator to incorporate information from each respective balanced and unbalanced sets. For example, in the first layer of Figure 2 we can see that the two positive neighbors of $u_i$ will be incorporated into the level one "friend" representation through the use of aggregator $B(1)$. Similarly, through the use of a second layer in our GCN, we can incorporate the two-hop neighbors. However, the crucial step here is that we must aggregate the information of these neighbors correctly to adhere to balance theory. Therefore we employ a second set of aggregators, namely $B(2)$ and $U(2)$ which will help propagate the information from users in sets $B_i(2)$ and $U_i(2)$, respectively. For example, aggregator $U(2)$ seeks to utilize the information from users along paths that consist of one positive and one negative link (in either ordering).

While aggregating and propagating information in our SGCN, we will maintain two representations at each layer, one for the corresponding balanced set of users (i.e., suggested "friends"), and one for the users in the respective unbalanced set (i.e., suggested "foes"). Similar to the unsigned GCN, we use $\mathbf{h}_i^{(0)} \in \mathbb{R}^{d^{in}}$ to represent the initial $d^{in}$ node features for user $u_i$. Thus, for the first aggregation layer, we utilize the following:

$$\mathbf{h}_i^{B(1)} = \sigma\left(\mathbf{W}^{B(1)}\left[\sum_{j \in \mathcal{N}_i^+} \frac{\mathbf{h}_j^{(0)}}{|\mathcal{N}_i^+|}, \mathbf{h}_i^{(0)}\right]\right) \tag{2}$$

$$\mathbf{h}_i^{U(1)} = \sigma\left(\mathbf{W}^{U(1)}\left[\sum_{k \in \mathcal{N}_i^-} \frac{\mathbf{h}_k^{(0)}}{|\mathcal{N}_i^-|}, \mathbf{h}_i^{(0)}\right]\right) \tag{3}$$

where $\mathbf{W}^{B(1)}, \mathbf{W}^{U(1)} \in \mathbb{R}^{d^{out} \times 2d^{in}}$ are for the "friends" and "foes" coming from sets $B_i(1)$ and $U_i(1)$, respectively, $\sigma()$ is a non-linear activation function, and $d^{out}$ is the length of the two internal hidden representations. More specifically, for determining the hidden representation $\mathbf{h}_i^{B(1)}$ we also concatenate the hidden representation of user $u_i$ along with the mean of the users in set $B_i(1)$. In all subsequent layers, the aggregation is more complex, just as the definition of $B_i(l)$ an $U_i(l)$ were more complex when $l > 1$ in (1) due to the cross linking of negative links as seen in Figure 1. The aggregations for $l > 1$ are defined as follows:

$$\mathbf{h}_i^{B(l)} = \sigma\Bigg(\mathbf{W}^{B(l)}\Big[\sum_{j \in \mathcal{N}_i^+} \frac{\mathbf{h}_j^{B(l-1)}}{|\mathcal{N}_i^+|}, \sum_{k \in \mathcal{N}_i^-} \frac{\mathbf{h}_k^{U(l-1)}}{|\mathcal{N}_i^-|}, \mathbf{h}_i^{B(l-1)}\Big]\Bigg)$$
$$\mathbf{h}_i^{U(l)} = \sigma\Bigg(\mathbf{W}^{U(l)}\Big[\sum_{j \in \mathcal{N}_i^+} \frac{\mathbf{h}_j^{U(l-1)}}{|\mathcal{N}_i^+|}, \sum_{k \in \mathcal{N}_i^-} \frac{\mathbf{h}_k^{B(l-1)}}{|\mathcal{N}_i^-|}, \mathbf{h}_i^{U(l-1)}\Big]\Bigg) \quad (4)$$

where $\mathbf{W}^{B(l)}, \mathbf{W}^{B(l)} \in \mathbb{R}^{d^{out} \times 3d^{out}}$ for $l > 1$. Note that we are utilizing the same logic here as when defining the sets $B_i(l)$ and $U_i(l)$. When gathering user $u_i$'s "friend" representation (i.e., $\mathbf{h}_i^{B(l)}$) at layer $l$ (when $l > 1$) it is based upon aggregating the "friend" representation at layer $(l-1)$ (i.e., $\mathbf{h}_k^{B(l-1)}$) for all positively linked neighbors $u_j \in \mathcal{N}_i^+$ while simultaneously collecting the average amongst the "foe" level $(l-1)$ (i.e, $\mathbf{h}_j^{U(l-1)}$) information from all negatively linked neighbors $u_k \in \mathcal{N}_i^-$. Thus, for the case when $l = 2$ we can see the "friend" representation is in fact gathering information from not only their direct friends (i.e., positively linked neighbors), but also (at the two hop level) friends of friends', and foes of foes'. Similarly, in the case of $l = 2$ our hidden representation $\mathbf{h}_i^{U(l)}$ (i.e., user $u_i$'s "foe" representation), the first layer would have gathered direct negatively linked neighbor information, but in the second layer, we are gathering from $u_i$'s friends' foes and their foes' friends.

We briefly summarize the entire framework of SGCN. The aggregation process in the first layer is performed according to (2) and (3). If the total number of layers in the SGCN is greater than one (i.e, $L > 1$), then we perform the subsequent aggregations according to (4). Finally, we concatenate the two hidden representations for user $u_i$, namely $\mathbf{h}_i^{B(L)}$ and $\mathbf{h}_i^{U(L)}$ together into a single low-dimensional representation.

Next we design an objective function to learn the parameters of SGCN. The objective function for SGCN is based upon two components. The first term incorporates an additional layer for performing a weighted multinomial logistic regression (MLG) classifier where we classify the relation between a pair of users. More specifically, we construct a set $\mathcal{M}$, which contains triplets of the form $(u_i, u_j, s)$ which denotes the pair of users ($u_i$,$u_j$) along with $s \in \{+, -, ?\}$ for denoting whether there was a positive, negative, or no link. For input into the classifier, we use the final embeddings for users $u_i$ and $u_j$ concatenated together (i.e,. $[\mathbf{z}_i, \mathbf{z}_j]$) and $\omega_s$ is the weight associated with class $s$. We introduce a second term that is founded on extended structural balance theory and controlled by $\lambda$. The goal of this second term is to have positively linked users closer in the embedded space than the no link pairs, and the no link paired users should be closer than users having a negative link between them. The overall objective is formalized in the following:

$$\begin{aligned}\mathcal{L}(\theta^W, \theta^{MLG}) &= Reg(\theta^W, \theta^{MLG}) \\ &- \frac{1}{\mathcal{M}} \sum_{(u_i,u_j,s) \in \mathcal{M}} \omega_s \log \frac{\exp\left([\mathbf{z}_i, \mathbf{z}_j]\theta_s^{MLG}\right)}{\sum_{q \in \{+,-,?\}} \exp\left([\mathbf{z}_i, \mathbf{z}_j]\theta_q^{MLG}\right)} \\ &+ \lambda\Bigg[\frac{1}{|\mathcal{M}_{(+,?)}|} \sum_{\substack{(u_i,u_j,u_k) \\ \in \mathcal{M}_{(+,?)}}} \max\left(0, (||\mathbf{z}_i - \mathbf{z}_j||_2^2 - ||\mathbf{z}_i - \mathbf{z}_k||_2^2)\right) \\ &+ \frac{1}{|\mathcal{M}_{(-,?)}|} \sum_{\substack{(u_i,u_j,u_k) \\ \in \mathcal{M}_{(-,?)}}} \max\left(0, (||\mathbf{z}_i - \mathbf{z}_k||_2^2 - ||\mathbf{z}_i - \mathbf{z}_j||_2^2)\right)\Bigg] \end{aligned} \quad (5)$$

$\theta^W$ represents the weight matrices used in the layers of our SGCN, $\theta^{MLG}$ denotes the parameters of the MLG classifier, $Reg(\theta^W, \theta^{MLG})$ we use for regularization, $\mathcal{M}_{(+,?)}$ and $\mathcal{M}_{(-,?)}$ are the sets for the pairs of positive and negatively linked users, respectively, where for every linked pair $(u_i, u_j)$ we further sample another user $u_k$ randomly having no link to $u_i$. For updating the parameters, we utilize the same stochastic gradient descent (SGD) style updating as presented in [6].

## III. Experiments

In this section, we experimentally evaluate the effectiveness of the proposed signed graph convolutional network (SGCN) in learning node representations. We seek to answer the following questions: (1) Is SGCN capable of learning meaningful low-dimensional representations? and (2) Does the introduction of balance theory into the aggregation process along with longer path information provide a performance increase in learning the node embeddings?

To address the first question, we conduct experiments to measure the learned embedding quality by performing the most fundamental signed network analysis task, namely link sign prediction [18]. To answer the second question, we investigate variants of our framework that do not exploit the longer paths or that do not make use of balance theory.

### A. Experimental Settings

In this subsection, we begin by introducing our datasets, the link sign prediction problem, and the metrics used for evaluation.

For our study of learning representations using signed graph convolutional networks, we conduct our experiments on four real-world signed network datasets, i.e., Bitcoin-Alpha, Bitcoin-OTC, Slashdot and Epinions. We note that for each of these datasets we perform our experiments on the undirected signed networks and have further filtered out users randomly from the two larger networks (Slashdot and Epinions) that had very few links. We summarize these datasets in Table I with some basic statistics.

The problem of predicting the signs of links [18] is that given a set of existing links in the signed network that had been held out of the training set, we wish to predict their signs between those pairs of users. Thus, a binary classifier is used

TABLE I
STATISTICS OF FOUR SIGNED SOCIAL NETWORKS.

| Network | # Users | # Positive Links | # Negative Links |
|---|---|---|---|
| Bitcoin-Alpha | 3,784 | 12,729 | 1,416 |
| Bitcoin-OTC | 5,901 | 18,390 | 3,132 |
| Slashdot | 33,586 | 295,201 | 100,802 |
| Epinions | 16,992 | 276,309 | 50,918 |

TABLE II
LINK SIGN PREDICTION RESULTS WITH (AUC,F1).

| Embedding Method | Bitcoin-Alpha | Bitcoin-OTC | Slashdot | Epinions |
|---|---|---|---|---|
| SSE | (0.764, 0.898) | (0.803, 0.923) | (0.769, 0.820) | (0.822, 0.901) |
| SiNE | (0.778, 0.888) | (0.814, 0.878) | (0.792, 0.854) | (0.849, 0.914) |
| SIDE | (0.630, 0.738) | (0.618, 0.750) | (0.547, 0.646) | (0.571, 0.711) |
| SGCN-1 | (0.780, 0.910) | (0.818, 0.918) | (0.784, 0.853) | (0.663, 0.851) |
| SGCN-1+ | (0.785, 0.912) | (0.817, 0.923) | (**0.804**, **0.865**) | (0.722, 0.893) |
| SGCN-2 | (**0.796**, **0.917** ) | (**0.823**, **0.925**) | (**0.804**, 0.864) | (**0.864**, **0.933**) |

to predict the sign based on a set of input features from the pair of users (more specifically we employ a logistic regression model). In our case we concatenate the final embeddings of the two users together as the set of features. The model is trained using the labeled edges from the training data. For evaluation, since the positive and negative links are unbalanced, we utilize both F1 score and Area Under the receiver operating characteristic Curve (AUC). We note that higher F1 and AUC both mean better performance. For each dataset, we randomly choose 20% of the data as test, and the remaining 80% as training. Note that we used a grid search along with cross validation on the training data to tune the hyperparameters of our model.

### B. Performance Comparison

Here we present some existing state-of-the-art signed network embeddings methods. For succinctness we do not include unsigned methods since previous signed network embedding work has shown their superiority over the non-dedicated efforts towards signed network embeddings. The baselines are as follows:

- Signed Spectral Embedding (SSE) [12]: A spectral clustering algorithm based on the proposed signed version of the Laplacian matrix. We utilize the top-$d^{out}$ eigenvectors corresponding to the smallest eigenvalues as the embedding vectors for each node.
- SiNE [19]: This method is a deep learning framework that utilized extended structural balance theory.
- SIDE [20]: A random walk based method, utilizing balance theory, is used to obtain indirect connections for a likelihood formulation.

Furthermore, we propose to evaluate the following two variants of our model:

- SGCN-1: This method only makes use of the first single aggregation layer and therefore only separates the positive from the negative links (i.e, does not yet make use of balance theory and our defined balanced paths).
- SGCN-1+: This method similar to SGCN-1 does not make use of balance theory, instead it performs the naïve aggregation of the first layer, but twice. In other words, the final representation for each user is based on propagating information along the positive links twice, and the negative links twice, separately.

Some final notes are the following: 1) in our experiments we do not have node attributes, therefore instead we use the final embedding of the SSE model as the input feature matrix to all our SGCN variants; 2) for all embedding methods we fixed the final low-dimensional representation to be 64; 3) we used the authors released code for SiNE[1] and use their suggested hyperparameters [19] for our experiments; 4) for SIDE, we use the authors implementation[2] and the suggested hyperparameter settings from [20], but for the unsuggested parameters we used a grid search around their code's default settings; 5) our code for SGCN-2 can be found on our website[3]; and 6) for our models we set $\lambda = 5$ and the "friend" and "foe" hidden representations were each set to 32, such that the final embeddings were of size 64.

### C. Comparison Results

The comparison results in terms of AUC and F1 are demonstrated in Tables II. For this table, we make the following observations: 1) SGCN-1 obtains comparable performance with the best performance from the baselines. This observation suggests that it is necessary to separate positive and negative links; 2) SGCN-1+ outperforms SGCN-1. The results indicate that propagating multiple steps during the aggregation can help improve the performance; and 3) Most of the time, SGCN-2 outperforms SGCN-1 and SGCN-1+. Aggregation following the longer balance and unbalanced paths can boost the performance.

## IV. RELATED WORK

In this section, we present and discuss related work on signed networks and on the recent development of graph convolutional networks.

While some problems such as link sign prediction [18], [21] and centrality [22] have been heavily explored, other directions such as network modeling [23], node relevance [24], and network embedding [19] are still in early development [25].

In [12] they extended spectral analysis and in [26] a matrix factorization approach was used to learn node embeddings in signed networks. Later, SiNE [19], which developed a deep learning framework, and SNE [27], that uses a log-bilinear model with random walks, were developed. Thereafter, works were introduced to handle the attributed [28] and directed settings [20]. Recently a signed heterogeneous network embedding algorithm, SHINE [29], had been developed, but in their work the social network itself is unsigned and thus leaving their method not applicable to the traditional signed network embedding problem.

[1] http://www.public.asu.edu/∼swang187/codes/SiNE.zip
[2] https://datalab.snu.ac.kr/side/resources/side.zip
[3] http://www.cse.msu.edu/∼derrtyle/

The early works on using convolutional neural networks for graphs focused on employing them on the entire network (and hence do not scale to larger networks) and/or designed for learning representations for the whole entire network for graph classification [1], [3], [5], [7] (as compared to learning node representations). The original GCN algorithm [4] was instead designed to learn representations at the node level. More recently, GraphSAGE [6] extended the ideas presented in [4] by first providing an inductive setting, introduced numerous aggregation functions, and an efficient strategy to utilize SGD. However, all the above mentioned GCNs have only been defined to handle unsigned networks.

## V. Conclusion

In conclusion, since existing GCNs are inherently unable to handle the complexities and challenges associated with the inclusion of negative links, we proposed a dedicated effort using balance theory, the fundamental social psychology theory designed to provide insights into how positive and negative links interact in the complex signed networks. This allowed us to bridge the gap between the recent advances in unsigned GCNs and the domain of signed network analysis. Using our constructed signed graph convolutional network, we performed empirical evaluations through experiments on four real-world signed networks showing the superiority of SGCNs over state-of-the-art signed network embedding methods for link sign prediction.

For our future work, we first plan to further investigate the usage of SGCNs for other signed networks tasks, such as node classification and tie strength predictions. Thereafter, we will focus on using other deep learning architectures for constructing a deep generative network model for signed networks.

## VI. Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work is supported by the National Science Foundation (NSF) under grant number IIS-1714741 and IIS-1715940.

## References

[1] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International conference on machine learning*, 2016, pp. 2014–2023.

[2] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*. Springer, 2018, pp. 593–607.

[3] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016.

[5] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[6] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1025–1035.

[7] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations (ICLR), CBLS, April 2014*, 2014.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] S. Wang, J. Tang, C. Aggarwal, and H. Liu, "Linked document embedding for classification," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 115–124.

[10] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 855–864.

[11] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding." in *AAAI*, 2017, pp. 7203–209.

[12] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak, "Spectral analysis of signed graphs for clustering, prediction and visualization," in *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 2010, pp. 559–570.

[13] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[14] 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.

[15] F. Heider, "Attitudes and cognitive organization," *The Journal of psychology*, vol. 21, no. 1, pp. 107–112, 1946.

[16] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory." *Psychological review*, vol. 63, no. 5, p. 277, 1956.

[17] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 1361–1370.

[18] ——, "Predicting positive and negative links in online social networks," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 641–650.

[19] S. Wang, J. Tang, C. Aggarwal, Y. Chang, and H. Liu, "Signed network embedding in social media," in *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 2017, pp. 327–335.

[20] J.-E. L. Junghwan Kim, Haekyu Park and U. Kang, "Side: Representation learning in signed directed networks," in *Proceedings of the 27th International Conference on World Wide Web (The Web Conference)*, 2018.

[21] T. Derr, Z. Wang, and J. Tang, "Opinions power opinions: Joint link and interaction polarity predictions in signed networks," in *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2018*, ser. ASONAM '18. ACM, 2018.

[22] J. Kunegis, A. Lommatzsch, and C. Bauckhage, "The slashdot zoo: mining a social network with negative edges," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 741–750.

[23] T. Derr, C. Aggarwal, and J. Tang, "Signed network modeling based on structural balance theory," in *Proceedings of the 2018 ACM on Conference on Information and Knowledge Management*. ACM, 2018.

[24] T. Derr, C. Wang, S. Wang, and J. Tang, "Relevance measurements in online signed social networks," in *Proceedings of the 14th International Workshop on Mining and Learning with Graphs (MLG)*, 2018.

[25] J. Tang, Y. Chang, C. Aggarwal, and H. Liu, "A survey of signed network mining in social media," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 42, 2016.

[26] K.-Y. Chiang, J. J. Whang, and I. S. Dhillon, "Scalable clustering of signed networks using balance normalized cut," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 615–624.

[27] S. Yuan, X. Wu, and Y. Xiang, "Sne: Signed network embedding," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2017, pp. 183–195.

[28] S. Wang, C. Aggarwal, J. Tang, and H. Liu, "Attributed signed network embedding," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 137–146.

[29] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "Shine: Signed heterogeneous information network embedding for sentiment link prediction," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 592–600.