Signed Network Embedding in Social Media

Suhang Wang* Jiliang Tang† Charu Aggarwal‡ Yi Chang§ Huan Liu*

Abstract

Network embedding is to learn low-dimensional vector representations for nodes of a given social network, facilitating many tasks in social network analysis such as link prediction. The vast majority of existing embedding algorithms are designed for unsigned social networks or social networks with only positive links. However, networks in social media could have both positive and negative links, and little work exists for signed social networks. From recent findings of signed network analysis, it is evident that negative links have distinct properties and added value besides positive links, which brings about both challenges and opportunities for signed network embedding. In this paper, we propose a deep learning framework SiNE for signed network embedding. The framework optimizes an objective function guided by social theories that provide a fundamental understanding of signed social networks. Experimental results on two real-world datasets of social media demonstrate the effectiveness of the proposed framework SiNE.

1 Introduction

The increasing availability of large-scale social media networks has greatly advanced social network analysis; and network embedding, which aims to learn low-dimensional vector representations for nodes, has been proven to be useful in many tasks of social network analysis such as link prediction [1], community detection [2], node classification/clustering [3, 4, 5] and visualization [6]. The vast majority of existing algorithms have been designed for social networks without sign or only with positive links. However, social networks can contain both positive and negative links, and such signed social networks are present on a variety of social media sites, such as Epinions with trust and distrust links, and Slashdot with friend and foe links. The work on signed network embedding is rather limited.

The availability of negative links in signed networks challenges some principles that explain the formation and properties of links for unsigned social networks; and principles for signed social networks can be substantially different from that of unsigned network [7, 8]. For example, homophily effects and social influence for unsigned networks may not be applicable to signed networks [9]. Therefore, signed network embedding cannot be carried out by simply extending embedding algorithms for unsigned social networks. Recent research on mining signed social networks suggests that negative links have added value over positive links in various analytical tasks. For example, a small number of negative links can significantly improve positive link prediction performance [10], and they can also improve recommendation performance in social media [11]. While signed network embedding is challenging, the results of such an approach have the potential to greatly advance tasks of mining signed social networks such as link prediction.

In this paper, we investigate the problem of signed network embedding in social media. To achieve this goal, we need (1) an objective function for signed network embedding since the objective functions of unsigned network embedding cannot be applied directly; and (2) a representation learning algorithm to optimize the objective function. Social theories distilled from social sciences provide fundamental understandings about signed social networks and have powered various tasks of mining signed social networks [12], while deep learning techniques provide powerful tools for representation learning which have enhanced various domains such as speech recognition, natural language processing and computer vision [13]. This motivates the proposed deep learning framework SiNE for signed network embedding. SiNE learns low-dimensional vector representations of nodes while preserves the fundamental understanding about signed social networks from a social theory. The major contributions of this paper are as follows:

- Design an objective function for signed social network embedding guided by social theories;
- Propose a deep learning framework SiNE for signed network embedding, which learns low-dimensional vector representations for nodes by optimizing the objective function; and
- Conduct experiments on two signed social networks from social media to demonstrate the effectiveness of the proposed framework SiNE.

The rest of the paper is organized as follows. In Section 2, we review related work. In Section 3, we introduce the proposed framework SiNE with the details

*Arizona State University, {suhang.wang, huan.liu}@asu.edu
†Michigan State University, jiliang.tang@cse.msu.edu
‡IBM T.J. Watson, charu.aggarwal@us.ibm.com
§Huawei Research America, yichang@acm.org

about the embedding objective function, the proposed framework SiNE and the training algorithm with time complexity analysis. In Section 4, we show empirical evaluation with discussion. In Section 5, we present the conclusion and future work.

2 Related Work

Network embedding or network representation learning is to learn low-dimensional vector representations for nodes of a given network. It has been proven to be useful in many tasks of network analysis such as link prediction [1], community detection [2], node classification [3] and visualization [6]. Data sparsity is the common problem faced by these tasks. To address the sparsity issue, network embedding encodes and represents each node in a unified low-dimensional space, which facilitates us to better understand the semantic relatedness and further alleviates the inconveniences caused by sparsity [14]. Network embedding has attracted increasing attention and various methods have been proposed for unsigned network embedding [15, 16, 1, 6, 14, 17]. For example, in [16], spectral analysis is performed on Laplacian matrix and the top-k eigenvectors are used as the representations of nodes of the network. Similarity scores measured by unsigned network analysis methods such as Adamic/Adar and Katz are used in [1] to represent nodes for unsigned link prediction; t-SNE [6] embeds the weighted unsigned network to low dimension for visualization by using stochastic neighbor embedding; DeepWalk [14] introduces the idea of Skip-gram, a word representation model in NLP, to learn node representations from random walk sequences in social networks; node2vec [18] extends DeepWalk by defining a flexible notion of a node's network neighborhood and designing a biased random walk procedure; HOPE [19] studies the network embedding for directed network.

However, the aforementioned algorithms are designed for unsigned network and don't take negative links into consideration, while negative links have been proven to have distinct properties and added value over positive links [10, 20]. The majority of network embedding algorithms ,such as spectral analysis, t-SNE, DeepWalk and node2vec, utilize the homophily effects or social influence that two linked nodes are likely to be similar and so as their vector representations. However, this is not true for signed network due to the existence of negative links, which are usually used to denote distrust or foe relationship between two nodes. Thus, signed network embedding cannot be carried out by simply extending embedding algorithms for unsigned social networks. In addition, negative links have added value over positive links in various analytical tasks. For example, a small number of negative links can significantly improve positive link prediction performance [10], and they can also improve recommendation performance in social media [11]. Though negative links are valuable, the work on signed network embedding is limited. In [10], degree based features such as the number of incoming positive and negative links of a node and triad based features that include the structure information of a triad are defined manually and extracted from the network to represent the nodes for sign prediction in signed network. Another work in [21] extends spectral analysis for signed network. In this paper, we study the novel problem of learning embedding for signed social network by utilizing social theory. In particular, we propose a novel framework SiNE, which models the extended structural balance theory and optimizes a deep network based objective function to learn signed network embedding automatically.

3 Signed Network Embedding

Before introducing details about the proposed framework, we first introduce the notations used in this paper. Throughout this paper, matrices are written in boldface capital letters and vectors are denoted as boldface lowercase letters. For an arbitrary matrix $\mathbf{M}$, $\mathbf{M}_{ij}$ denotes the (i,j)-th entry of $\mathbf{M}$ while $\mathbf{m}^i$ and $\mathbf{m}_j$ mean the i-th row and j-th column of $\mathbf{M}$, respectively. The i-th element of a vector $\mathbf{m}$ is denoted as m_i. $\|\mathbf{M}\|_F$ is the Frobenius norm of $\mathbf{M}$. The transpose of a matrix $\mathbf{M}$ and a vector $\mathbf{m}$ is denoted as $\mathbf{M}^T$ and $\mathbf{m}^T$, respectively. Capital letters in calligraphic math font such as $\mathcal{V}$ are used to denote sets and $|\mathcal{V}|$ is the cardinality of $\mathcal{V}$. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a signed network where $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$ is a set of m nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of links. Particularly, any link $e_{ij} \in \mathcal{E}$ can be 1 or -1 where $e_{ij} = 1$ denotes a positive link between v_i and v_j, while $e_{ij} = -1$ denotes a negative link. Next we will first introduce the objective function for signed network embedding, and then detail the deep learning framework SiNE and the training algorithm.

3.1 An Objective Function for Signed Network Embedding

Recent research about signed social networks suggests that negative links present distinct properties from positive links, and the fundamental principles that drive the formation of links for signed and unsigned social networks are very different [9, 22, 7]. This suggests that we need a new objective function for signed network embedding because we cannot apply those for unsigned social networks directly on signed social networks.

Social theories are developed by social scientists to explain social phenomenon in signed social networks and they provide fundamental understandings about signed social networks. Social theories have been widely exploited in various tasks of mining signed social networks

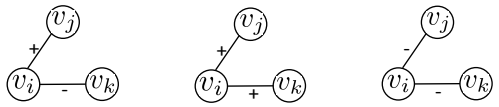


Figure 1: Three Types of Triplets of Users.

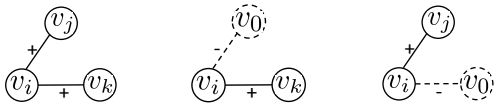


Figure 2: Adding a Virtual Node.

such as link prediction [7] and community detection [23]. The successful experiences on exploiting social theories in mining signed social networks suggest that social theories may guide us to develop objective functions for signed network embedding. Actually, social theories for unsigned social networks have been widely used to design objective functions for unsigned social network embedding. For example, social correlation theories such as homophily [24] and social influence [25] suggest that two connected users are likely to share similar interests, which are the foundations of many objective functions of unsigned network embedding [16, 15]. Inspired by the success of applying social theories in unsigned network, we seek social theories on signed network for signed network embedding. Among the social theories, structural balance theory is one of the most important and popular theories for signed social networks. Thus, in this work we develop an objective function for signed network embedding based on it.

Structural balance theory was originally proposed in [26] at the individual level, generalized by Cartwright and Harary [27] in the graph-theoretical formation at the group level and then was developed to the concept of clusterizable graph in [28]. It is recently extended by [29] as: a structure in signed social network should ensure that users should be able to have their "friends" closer than their "foes", i.e., users should sit closer to their "friends" (or users with positive links) than their "foes" (or users with negative links). In other words, the key idea of extended structural balance theory suggests that a user should be more similar to her friends than her foes. The extended structural balance theory provides us a guidance to model signed social network for learning network embedding. We will now introduce the detail of how we model signed social network based on extended structural balance theory.

Let $\mathcal{P}$ be a set of triplets (v_i, v_j, v_k) as shown in Figure 1(a) from a given signed social network $\mathcal{G}$, where v_i and v_j have a positive link while v_i and v_k have a negative link. Formally, $\mathcal{P}$ is defined as:

$$\mathcal{P} = \{(v_i, v_j, v_k) | e_{ij} = 1, \quad e_{ik} = -1, \quad v_i, v_j, v_k \in \mathcal{V}\},$$

The extended structural balance theory in [29] suggests that with a certain similarity measurement, for a triplet $(v_i, v_j, v_k) \in \mathcal{P}$, v_i is likely to be more similar to the user with a positive link, i.e. v_j, than a user with a negative link, i.e. v_k, which can be mathematically modeled as:

$$f(\mathbf{x}_i, \mathbf{x}_j) \geq f(\mathbf{x}_i, \mathbf{x}_k) + \delta, \tag{3.1}$$

where $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ are the d-dimensional vector representations of v_i, v_j and v_k respectively, which we need to learn by the proposed embedding framework. In $f(\mathbf{x}_i, \mathbf{x}_j)$, f is a function that measures the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. We will discuss more details about the function f in the proposed framework in the following subsection. The parameter δ is a threshold that is used to regulate the difference between these two similarities. A large δ will push v_i, v_j more close and v_i, v_k more far away. The range of δ will be discussed in experimental analysis section.

In a real-world signed network, the objective function in Eq. (3.1) has no effect on those nodes whose 2-hop networks[1] have only positive or negative links. That is to say, we cannot learn the d-dimensional vector representations for those nodes because there are no triplets in $\mathcal{P}$ that contains them. Those nodes are involved in triplets as shown in Figures 1(b) and Figures 1(c). According to a recent study [9], the cost of forming negative links is higher than that of forming positive links in social media. Therefore, in a signed social network, positive links are denser than negative links. This determines that there are many nodes whose 2-hop networks have only positive links while very few nodes whose 2-hop networks have only negative links. Therefore, next we only consider handling nodes whose 2-hop networks have only positive links although a similar solution can be applied to dealing with the other type of nodes.

We first introduce a virtual node v_0 and then create a negative link between v_0 and each node whose 2-hop network has only positive links. In this way, an original triplet (v_i, v_j, v_k) in Figure 2(a) (or Figure 1(b)) will lead to two triplets (or (v_i, v_j, v_0) and (v_i, v_k, v_0)) as shown in Figures 2(b) and 2(c). Let $\mathcal{P}_0$ be the set of triplets (v_i, v_j, v_0) where v_i and v_j have a positive link while v_i and v_0 have a negative link, and a similar objective function as Eq. (3.1) can be developed as:

$$f(\mathbf{x}_i, \mathbf{x}_j) \geq f(\mathbf{x}_i, \mathbf{x}_0) + \delta_0, \tag{3.2}$$

[1] v_i's 2-hop network is defined as the network formed by v_i, users whose distance from v_i is within 2 hops and links among them.

where δ_0 is a threshold to regulate the similarities. The reason of using δ_0 in Eq.(3.2) and δ in Eq.(3.1) is that we can have more flexibility to distinguish triplets with or without the virtual node by tuning δ and δ_0. By adding the virtual node, we can make a node v_i whose 2-hop network contains only positive links closer to their neighbors.

3.2 The Proposed Framework SiNE Based on Eq.(3.1) and (3.2), the objective function for signed social network embedding guided by the extended structural balance theory can be written as:

(3.3)
$$\min_{\mathbf{X},\mathbf{x}_0,\theta} \frac{1}{C}\Big[\sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{P}} \max(0, f(\mathbf{x}_i,\mathbf{x}_k)+\delta-f(\mathbf{x}_i,\mathbf{x}_j)) + \sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_0)\in\mathcal{P}_0} \max(0, f(\mathbf{x}_i,\mathbf{x}_0)+\delta_0-f(\mathbf{x}_i,\mathbf{x}_j))\Big] + \alpha\big(\mathfrak{R}(\theta)+\|\mathbf{X}\|_F^2+\|\mathbf{x}_0\|_2^2\big),$$

where $C = |\mathcal{P}| + |\mathcal{P}_0|$ is the size of the training data and $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ is the low-dimensional representation of the m nodes, and θ is a set of parameters to define the similarity function f. $\mathfrak{R}(\theta)$ is the regularizer to avoid overfitting and α is a parameter to control the contribution of the regularizers.

3.3 The Architecture of SiNE With the objective function given above, the task now is to find a function f that is able to give good similarity measure and learn good representations of nodes in signed network. One choice of f is non-linear functions, which have shown to be superior than linear functions for similarity measure and representation learning [30]. Among various non-linear functions, deep learning has been proven to be the state-of-the-art and very powerful for nonlinear representation learning [31, 30]. This suggests us to utilize the power of deep learning for learning nonlinear embedding of the nodes. In particular, we design a deep learning framework SiNE, which defines f with θ and optimizes the objective function in Eq. (3.3). To help better understand SiNE, we first work on an illustrative example of the architecture of the proposed deep learning framework with 2 hidden layers (see Figure 3) and then generalize it to N layers. Note that we do not show bias in the figure. The input to the framework is the set of triplets extracted from the signed social network as (v_i, v_j, v_k) with $e_{ij} = 1$ and $e_{ik} = -1$. The model is composed of two deep networks that share the same parameters. The outputs of the first hidden layer of the two deep networks (or "1st Hidden Layer " in Figure 3) are given as:

(3.4)
$$\mathbf{z}^{11} = \tanh(\mathbf{W}^{11}\mathbf{x}_i + \mathbf{W}^{12}\mathbf{x}_j + \mathbf{b}^1)$$
$$\mathbf{z}^{12} = \tanh(\mathbf{W}^{11}\mathbf{x}_i + \mathbf{W}^{12}\mathbf{x}_k + \mathbf{b}^1)$$

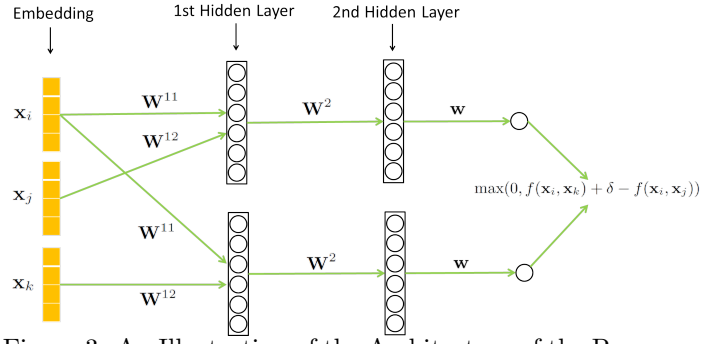


Figure 3: An Illustration of the Architecture of the Proposed Deep Learning Framework SiNE with 2 Hidden Layers.

where *tanh* is the hyperbolic tangent function, which is one of the most widely used activation function in deep networks. $\mathbf{W}^{11}$ and $\mathbf{W}^{12}$ are the weights of the first hidden layer and $\mathbf{b}^1$ is the bias. $\mathbf{z}^{11}$ and $\mathbf{z}^{12}$ are then used as inputs to the second hidden layer (or "2nd Hidden Layer" in Figure 3) of the two deep networks, separately. Similarly, the outputs of the second layer are $\mathbf{z}^{21} = \tanh(\mathbf{W}^2\mathbf{z}^{11} + \mathbf{b}^2)$ and $\mathbf{z}^{22} = \tanh(\mathbf{W}^2\mathbf{z}^{12} + \mathbf{b}^2)$. $f(\mathbf{x}_i, \mathbf{x}_j)$ and $f(\mathbf{x}_i, \mathbf{x}_k)$ are the output of the two deep networks:

(3.5)
$$f(\mathbf{x}_i,\mathbf{x}_j) = \tanh(\mathbf{w}^T\mathbf{z}^{21} + b)$$
$$f(\mathbf{x}_i,\mathbf{x}_k) = \tanh(\mathbf{w}^T\mathbf{z}^{22} + b)$$

which are the terms in Eq. (3.3) and the vector $\mathbf{w}$ is the weights and the scalar b is the bias[2]. With the illustration of the proposed framework with 2 hidden layers, we can see that the similarity function f is defined by the deep network with a set of parameters as shown in Figure 3. Particularly, in Figure 3, θ is defined as $\theta = \{\mathbf{W}^{11}, \mathbf{W}^{12}, \mathbf{W}^2, \mathbf{w}, \mathbf{b}^1, \mathbf{b}^2, b\}$ and correspondingly we define $\mathfrak{R}(\theta)$ as:

(3.6)
$$\mathfrak{R}(\theta) = \|\mathbf{W}^{11}\|_F^2 + \|\mathbf{W}^{12}\|_F^2 + \|\mathbf{W}^2\|_2^2 + \|\mathbf{w}\|_2^2 + \|\mathbf{b}^1\|_2^2 + \|\mathbf{b}^2\|_2^2 + b^2$$

Note that we can also choose other regularizers for θ such as those based on ℓ_1-norm and we would like to leave it as one future work.

We now extend the 2 hidden layer example to a N layer deep network. For a N layer deep network, the parameters are $\mathbf{X}$, $\mathbf{x}_0$ and $\theta = \{\mathbf{W}^{11}, \mathbf{W}^{12}, \mathbf{W}^2, \ldots, \mathbf{W}^N, \mathbf{b}^1, \ldots, \mathbf{b}^N, \mathbf{w}, b\}$ where $\mathbf{W}^n$ are the weights for the n-th layer and $\mathbf{b}^n$ is the bias for n-th layer with $1 < n \leq N$. The input to the first hidden layer is triplet (v_i, v_j, v_k), i.e., $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$. And the input to the n-th layer, $1 < n \leq N$, is the output

[2]Note that for the proposed framework SiNE, the weights and bias to generate output $f(\mathbf{x}_i, \mathbf{x}_j)$ are a vector and a scalar, separately

of the $(n-1)$-th layer, i.e., $\mathbf{z}^{(n-1)1}$ and $\mathbf{z}^{(n-1)2}$. The output of the first layer is given by Eq. (3.4) and the output of the n-th layer, $1 < n < N$ is given as:

$$\begin{aligned} \mathbf{z}^{n1} &= \tanh(\mathbf{W}^n \mathbf{z}^{(n-1)1} + \mathbf{b}^n) \\ \mathbf{z}^{n2} &= \tanh(\mathbf{W}^n \mathbf{z}^{(n-1)2} + \mathbf{b}^n) \end{aligned} \tag{3.7}$$

And the output of the N-th layer is given as

$$\begin{aligned} f(\mathbf{x}_i, \mathbf{x}_j) &= \tanh(\mathbf{w}^T \mathbf{z}^{N1} + b) \\ f(\mathbf{x}_i, \mathbf{x}_k) &= \tanh(\mathbf{w}^T \mathbf{z}^{N2} + b) \end{aligned} \tag{3.8}$$

3.4 Optimization of SiNE Following the common way, we employ the backpropagation to optimize the deep network for SiNE. The key idea of backpropagation is to update the parameters in a backward direction by propagating "errors" backward to efficiently calculate the gradients. Basically, we want to optimize Eq. (3.3) w.r.t to $\mathbf{X}$, $\mathbf{x}_0$ and θ. The key step of optimizing Eq. (3.3) is to get the gradient of $\max(0, f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j))$ and $\max(0, f(\mathbf{x}_i, \mathbf{x}_0) + \delta - f(\mathbf{x}_i, \mathbf{x}_j))$ with respect to the parameters, $\mathbf{X}, \mathbf{x}_0$ and θ. With the gradient, we then can update the parameters using gradient descent method. The details of how to derive the derivatives for backpropagation can be found in Appendix [3].

3.5 Training SiNE We train SiNE based on mini-batch stochastic gradient descent with respect to the parameters of the deep network, i.e. θ, the signed network embedding $\mathbf{X}$ and the virtual node embedding $\mathbf{x}_0$. It is well known that for signed social networks in social media, the number of links of nodes follows power-law distributions, i.e., many nodes have only a small number of links while only a small number of nodes have a large number of links. This will cause some nodes to have a large number of training triplets. To save computational cost, following the same idea used in word embedding [32], for a node that has a large number of training triplets, we randomly sample a subset of the training triplets for training. The size of the subset is chosen as $S = 300$. In other words, each node has at most 300 training triplets. The initialization of the parameters of the deep network follows the approach introduced in [33]. Specifically, we initialize the weights of hidden layer i by a uniform sampling from the interval $\big[-\sqrt{(\frac{6}{d_{i-1}+d_i})}, \sqrt{(\frac{6}{d_{i-1}+d_i})}\big]$, where d_{i-1} is number of units in the $(i-1)$-th layer and d_i is the number of units in the i-th layer. The signed network embedding $\mathbf{X}$ is initialized as a zero matrix. The training algorithm for the proposed framework SiNE is summarized in Algorithm 1. From line 1 to line 9, we prepare the mini-batch training triplets. In line 10, we initialize the parameters of the deep network and the low-dimensional representations and we train the deep network from line 11 to line 24.

[3] Available on http://www.public.asu.edu/ swang187/

Algorithm 1 Signed Network Embedding

Require: Signed social network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, d, δ, α
Ensure: vector representation of nodes $\mathbf{X}$
1: Initialize $\mathcal{P}$ and $\mathcal{P}_0$ as $\mathcal{P} = \emptyset$ and $\mathcal{P}_0 = \emptyset$
2: **for** i=1:n **do**
3: **if** v_i whose 2-hop networks have only positive links **then**
4: extract triplets with virtual nodes and put them into $\mathcal{P}_0$ (sample some if necessary)
5: **else**
6: extract triplets and put them in $\mathcal{P}$ (sample some if necessary)
7: **end if**
8: **end for**
9: prepare mini-batch from $\mathcal{P}$ and $\mathcal{P}_0$
10: initialize the parameters of the deep network and signed network embedding
11: **repeat**
12: **for** each mini-batch **do**
13: *Forward propagation*
14: **for** n = 1:N **do**
15: calculate $\mathbf{z}^{n1}, \mathbf{z}^{n2}$
16: **end for**
17: *Backpropagation*
18: Update $\mathbf{w}$ and b
19: **for** n= N:1 **do**
20: update $\mathbf{W}^n$, $\mathbf{b}^n$ (or $\mathbf{W}^{11}, \mathbf{W}^{12}$ if $n = 1$)
21: **end for**
22: update related $\mathbf{X}$ and $\mathbf{x}_0$
23: **end for**
24: **until** Convergence
25: return $\mathbf{X}$

3.6 Time Complexity Let d be the dimension of the embedding and $d_n, 1 \le n \le N$, be the number of nodes in the n-th layer of the deep network. For a triplet, the computational cost of forward propagation in the n-th layer, i.e., the computation of $\mathbf{z}^{n1}, \mathbf{z}^{n2}$, is $\mathcal{O}(d_{i-1}d_i)$ and the computational cost of back-propatgation in the n-th layer, i.e. the computation of $\frac{\partial f(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{W}^n}$, is also $\mathcal{O}(d_{i-1}d_i)$. Thus, the cost of forward and backward propagation for one triplet is $\mathcal{O}(dd_1 + \sum_{n=1}^{N} d_{i-1}d_i)$. Since for each node, we sample no more than $S = 300$ training triplets, the total number of triplets for training, i.e., $C = |\mathcal{P}| + |\mathcal{P}|_0$, is approximately $\mathcal{O}(m \cdot S)$, where m is number of nodes. Thus, the overall computational cost for training SiNE is $\mathcal{O}(tmS(dd_1 + \sum_{n=1}^{N} d_{i-1}d_i))$, where t is number of epochs it takes to

converge. And in our experiments, t is about 100 for the datasets used.

4 Experimental Results

In this section, we conduct experiments to evaluate the effectiveness of the proposed framework SiNE and factors that could affect the performance of SiNE. We begin by introducing datasets. We then analyze the embedding learned by SiNE. To measure the quality of the embedding, we use the embedding for signed link perdition with comparisons with state-of-the-art baseline methods. Further experiments are conducted to study the effects of the dimension of the embedding on SiNE, parameters δ and δ_0 and number of layers N.

4.1 Datasets The experiments are conducted on two real-world signed social network datasets, i.e., Epinions[4] and Slashdot[5]. Epinions is a popular product review site in which users can create both trust (positive) and distrust (negative) links to other users. Slashdot is a technology news platform where users can create friend (positive) and foe (negative) links to other users. For both datasets, we filter out users who have no links, which leaves us 27,215 users for Epinions dataset and 33,407 users for Slashdot dataset. Some key statistics of the two datasets are summarized in Table 1. It is evident from the table that (1) both networks are very sparse; (2) positive links are denser than negative links.

Table 1: Statistics of the Datasets

Dataset	# users	# pos links	# neg links
Epinions	27,215	326,909	58,695
Slashdot	33,407	477,176	158,104

4.2 Analysis of the Embedding In this subsection, we would like to check whether the embedding learned by the deep learning framework SiNE can preserve the principle suggested by the extended structural balance theory - users are likely to be more similar to their friends than their foes. Specifically, we first train the model on the two datasets and learn the signed network embedding. In the experiment, we set d as 20, $\alpha = 0.0001$ and $N = 3$ with all the hidden layer dimension as 20. We will discuss the effects of d and N in detail later. Since we use *tanh* as the activation function whose range is $(1, -1)$, from Eq.(3.8), we have that $f(\mathbf{x}_i, \mathbf{x}_j) \in (-1, 1)$ and $f(\mathbf{x}_i, \mathbf{x}_k) \in (-1, 1)$. In order to let $f(\mathbf{x}_i, \mathbf{x}_j) \geq f(\mathbf{x}_i, \mathbf{x}_k) + \delta$ be valid, δ should be within the range $(0, 2)$. The same holds for δ_0. We empirically set $\delta = 1$ and $\delta_0 = 0.5$. Then for each triplet (v_i, v_j, v_k) where v_i and v_j have a positive link while v_i and v_k have a negative link, we calculate the Euclidean distance for pairs of (v_i, v_j) and (v_i, v_k) and the average Euclidean distance (with standard deviation) are demonstrated in Table 2. From the table, we note that after embedding, nodes are indeed closer to their friends (positive link) than their foes (negative link), which suggests that the embedding from SiNE can perverse the principle suggested by the extended structural balance theory.

Table 2: Average Distance between Users and Their Friends and Foes.

Dataset	Dis. to Friends (+)	Dis. to Foes (-)
Epinions	0.0584±0.0275	0.1195±0.0335
Slashdot	0.0538±0.0245	0.1028±0.0254

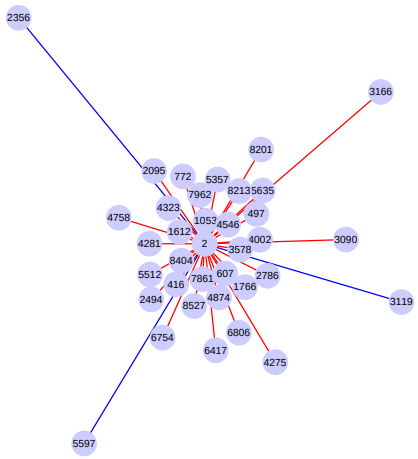


Figure 4: A Case Study of Signed Network Embedding. Note that red lines denote positive links and blue lines denote negative links.

A case study of the embedding distance between a user (or node 2) and his/her friends and foes is shown in Figure 4. The red lines denote positive links and the blue lines denote negative links. We use the length of the line to represent the embedding distance of two nodes. The longer the line is, the larger the embedding distance is. We observe from the figure that node 2 is likely to be closer to his/her friends than his/her foes.

4.3 Signed Link Prediction in Signed Social Networks The learned signed network embedding can benefit various mining tasks of signed social networks. In this subsection, we check whether the learned signed network embedding can improve the performance of link prediction for signed social networks. For both datasets, we randomly select 80% links as training set and the remaining 20% as test set. We use the training set to learn the signed network embedding. With the learned signed network embedding, we train a logistic regression classifier on training dataset. Then we predict link

[4]http://www.epinions.com/
[5]http://slashdot.org/

Table 3: AUC Comparison of Signed Link Prediction on Epinions and Slashdot

Dataset	SC	FExtra	MF	SiNE/$\mathcal{P}_0$	SiNE
Epinions	0.8527	0.8626	0.8879	0.8845	**0.9242**
Slashdot	0.8495	0.8536	0.8725	0.8701	**0.8979**

Table 4: F1 Comparison of Signed Link Prediction on Epinions and Slashdot

Dataset	SC	FExtra	MF	SiNE/$\mathcal{P}_0$	SiNE
Epinions	0.9089	0.9178	0.9343	0.9306	**0.9622**
Slashdot	0.8792	0.8839	0.8952	0.8924	**0.9149**

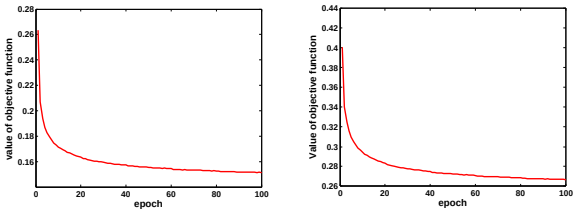
on the test set with the logistic regression classifier. In real-world signed social networks such as Epinions and Slashdot, positive links are often much denser than negative links; hence positive and negative links are imbalanced in both training and testing sets. Therefore, following the common way to evaluate the signed link prediction problem [10, 34], we use AUC and F1 instead of accuracy to assess the performance. The random selection is carried out 5 times independently and the average AUC and F1 are reported in Table 3 and 4. The baseline methods in the tables are defined as:

- SC [21]: A spectral clustering algorithm is proposed where a signed version of Laplacian matrix is defined. In this experiment, for the link prediction purpose, we choose the top-d eigen-vetors corresponding to the smallest eigenvalues of the signed Laplacian matrix as the low dimensional vector representations of nodes.
- FExtra [10]: This method extracts features from signed social networks. For each pair (v_i, v_j), the extracted features include degree based and triad based features. Degree based features contain the degree information such as the number of incoming positive and negative links of v_i, the number of outgoing positive and negative links of v_j and so on. Triad based features include the structure information of the triad that contains v_i and v_j.
- MF [35]: Matrix factorization based method which factorizes the adjacency matrix into two low rank latent matrices and predicts the links by the matrix reconstructed by the two low rank matrices.
- SiNE/$\mathcal{P}_0$: a variant of the proposed framework SiNE without considering virtual nodes. In other words, for SiNE/$\mathcal{P}_0$, we set $\mathcal{P}_0 = \emptyset$ in Eq.(3.3).

For SC, FExtra and the proposed framework SiNE, we first obtain the new representations and then choose logistic regression as the basic classifier for a fair comparison. 5-fold cross validation is performed on the training set to select the parameters for SC, FExtra and MF. For SiNE, we empirically set $d = 20$ $\delta = 1, \delta_0 = 0.5$ and $N = 3$ with all hidden layer dimension as 20. More details about parameters of SiNE will be discussed in the following subsection. From the Table 3 and 4, we make the following observations:

- The performance of the proposed framework SiNE is much better than FExtra. FExtra uses feature engineering to extract features manually; while SiNE learns the representations from the data automatically. These results suggest that the representations learned by SiNE can greatly improve the performance of link prediction; and
- The performance of SiNE outperforms SC and MF. SC designs a signed Laplacian to preserve pair-wise relations of nodes, while SiNE preserves the principle suggested by the extended structural balance theory, which supports the capability of the objective function of signed network embedding.
- SiNE outperforms SiNE/$\mathcal{P}_0$ because without considering the virtual node, the representation of nodes whose 2-hop networks have only negative links cannot be well trained. Since such nodes are few, the performance decrease is not much without considering virtual node.

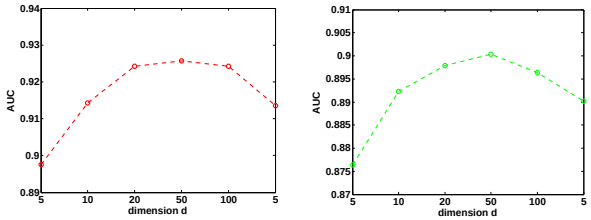
We perform t-test on comparisons and it is evident from t-test that the improvement of SiNE compared to baseline methods is significant. In summery, the representations learned by the proposed framework SiNE can significantly improve the performance of link prediction in signed social networks.



(a) Epinions (b) Slashdot

Figure 5: Convergence of SiNE on Epinions & Slashdot

4.4 Convergence Analysis The objective function of SiNE in Eq.(3.3) is non-negative. Thus, by minimizing it using gradient descent and choosing a proper learning rate, the objective function will reach a local optimal point and converge. We plot the value of the objective function in each training epoch for both Epinions and Slashdot in Figure 5. From the figure, we can see that the value of the objective function decreases fast at the first 10 iterations and then gradually converges. For both datasets, it takes about 100 epochs to converge.

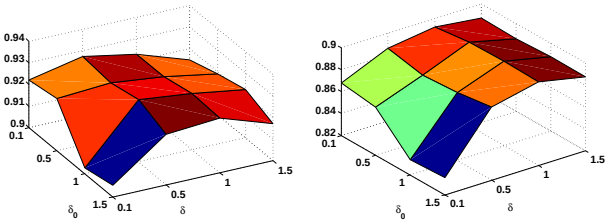


(a) AUC on Epinions (b) AUC on Slashdot

Figure 6: The Impact of Embedding Dimension d on SiNE for Signed Link Prediction

4.5 Parameter Analysis In this subsection, we investigate the impact of embedding dimesnion d, δ, δ_0 and number of layers N on the performance of link prediction. Throughout the experiments for parameter sensitively analysis, we randomly select 80% links as training set and the remaining 20% as test set. The random selection is repeated 5 times and the average AUC will be reported.

4.5.1 Impact of d: To investigate the sensitivity of SiNE on d, we fix $\delta = 1$, $\delta_0 = 0.5$ and $N = 3$. We then vary d as $\{5, 10, 20, 50, 100, 200\}$. The average AUC for signed link prediction on both datasets are shown in figure 6(a) and 6(b), respectively. From the two figures, we note that with the increase of d, the signed link prediction performance first increases and then decreases after certain values. When d is small, we may lose too much information and embeddings do not have enough representation capacity. When d is large, the embedding tends to overfit. A value of d around 20 gives relatively good performance, which eases the parameter selection for d.



(a) AUC on Epinions (b) AUC on Slashdot

Figure 7: The Impact of δ and δ_0 on SiNE for Signed Link Prediction.

4.5.2 Impact of δ and δ_0: As shown in Eq.(3.3), δ and δ_0 controls the similarity of a node with its friend and the node with its foe. To investigate the impact of δ and δ_0, we fix the dimension d to be 20 and $N = 3$. We then vary both δ and δ_0 as $\{0.1, 0.5, 1, 1.5\}$. The results in terms of AUC under different combinations of δ and δ_0 are shown in Figure 7. From the figure, we note that: (i) As the increase of δ, the performance generally increases. This is because when δ is large, we enforce a friend to sit closer to its friends and sit more far away from his foes, which help us to learn high quality embedding for signed link prediction; and (ii) When δ_0 is large and δ is small, e.g., $\delta_0 = 1.5$ and $\delta = 0.1$, the performance is relatively bad. A combination of (δ_0, δ) chosen from $[0.5, 1]$ generally result in good embedding for signed link prediction.

Table 5: AUC of SiNE on Signed Link Prediction with Different Number of Layers N

Dataset	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$
Epinions	0.9124	0.9242	0.9278	0.9297	0.9254
Slashdot	0.8817	0.8979	0.9048	0.9044	0.9027

4.5.3 Impact of N: To investigate the effects of N, we first fix d to be 20, δ to be 1 and δ_0 to be 0.5. We then vary N as 2, 3, 4, 5, 6 with all the hidden dimensions as 20. The results in terms of AUC are reported in Table 5. From the table, we can see that as the network becomes deeper, the performance increases first then the increase become small, which suggests that by setting $N = 2$ or $N = 3$, we can learn a relatively good embedding and at the same time save computational cost.

5 Conclusion

Most of the existing network embedding algorithms are designed for unsigned networks; while little work exists for signed social networks. Though signed network embedding is inherently challenging due to the availability of negative links, it can benefit various tasks of mining signed social networks. In this paper, we study the problem of signed network embedding. In particular, we introduce a new objective function for signed network embedding guided by extended structural balance theory and propose a deep learning framework SiNE to optimize this objective function. Via experiments on two signed networks in social media, we demonstrate (1) the learned embedding can preserve the principle of signed social networks indicated by extended structural balance theory; and (2) the embedding learned by SiNE can significantly improve the link prediction performance compared to representative baseline methods.

There are several directions needing further investigation. First, the proposed framework SiNE can only deal with undirected signed social networks and we would like to investigate how to extend it for directed signed social networks. Second, in this work we demonstrate that the embedding can improve the performance of link prediction in signed social networks; hence we will investigate how the embedding benefits other signed network mining tasks such as node classification, sentiment analysis [36] and multitask learning [37]. Finally, in addition to the structural balance theory, there are

other social theories for signed social network such as status theory; hence in the future, we will investigate how to develop objective functions for signed network embedding based on those social theories.

6 Acknowledgements

This material is based upon work supported by, or in part by, the NSF grants #1614576 and IIS-1217466, and the ONR grant N00014-16-1-2257.

References

[1] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[2] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *DMKD*, vol. 24, no. 3, pp. 515–554, 2012.

[3] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," in *Social network data analytics*. Springer, 2011, pp. 115–148.

[4] S. Wang, J. Tang, F. Morstatter, and H. Liu, "Paired restricted boltzmann machine for linked data," in *CIKM*. ACM, 2016, pp. 1753–1762.

[5] S. Wang, J. Tang, C. Aggarwal, and H. Liu, "Linked document embedding for classification," in *CIKM*. ACM, 2016, pp. 115–124.

[6] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, no. 2579-2605, p. 85, 2008.

[7] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *SIGCHI*, 2010.

[8] J. Tang, Y. Chang, C. Aggarwal, and H. Liu, "A survey of signed network mining in social media," *ACM Comput. Surv.*, vol. 49, no. 3, 2016.

[9] J. Tang, X. Hu, and H. Liu, "Is distrust the negation of trust?: the value of distrust in social media," in *Hypertext*. ACM, 2014, pp. 148–157.

[10] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *WWW*. ACM, 2010, pp. 641–650.

[11] H. Ma, M. R. Lyu, and I. King, "Learning to recommend with trust and distrust relationships," in *RecSys*. ACM, 2009, pp. 189–196.

[12] J. Tang, Y. Chang, and H. Liu, "Mining social media with social theories: A survey," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 2, pp. 20–29, 2014.

[13] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPRW*. IEEE, 2014.

[14] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *SIGKDD*. ACM, 2014, pp. 701–710.

[15] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[16] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *NIPS*, vol. 14, 2001, pp. 585–591.

[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*. ACM, 2015, pp. 1067–1077.

[18] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for network," in *SIGKDD*. ACM, 2016.

[19] M. Ou, P. Cui, J. Pei, and W. Zhu, "Asymmetric transitive preserving graph embedding," in *KDD*, 2016.

[20] J. Tang, S. Chang, C. Aggarwal, and H. Liu, "Negative link prediction in social media," in *WSDM*, 2015.

[21] 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 *SDM*, vol. 10. SIAM, 2010, pp. 559–559.

[22] 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*, vol. 107, no. 31, 2010.

[23] K.-Y. Chiang, J. J. Whang, and I. S. Dhillon, "Scalable clustering of signed networks using balance normalized cut," in *SIGIR*. ACM, 2012, pp. 615–624.

[24] S. Wasserman, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[25] J. C. Turner, *Social influence*. Thomson Brooks/Cole Publishing Co, 1991.

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

[27] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory." p. 277, 1956.

[28] J. A. Davis, "Clustering and structural balance in graphs," *Social networks. A developing paradigm*, pp. 27–34, 1977.

[29] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk, "Sitting closer to friends than enemies, revisited," in *Mathematical Foundations of Computer Science*. Springer, 2012, pp. 296–307.

[30] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *TPAMI*, vol. 35, no. 8, pp. 1798–1828, 2013.

[31] R. Socher, A. Perelygin, J. Y. Wu *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP*, 2013.

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010, pp. 249–256.

[34] G. Beigi, J. Tang, S. Wang, and H. Liu, "Exploiting emotional information for trust/distrust prediction," in *SDM*, 2016.

[35] C.-J. Hsieh, K.-Y. Chiang, and I. S. Dhillon, "Low rank modeling of signed networks," in *SIGKDD*, 2012.

[36] Y. Wang, S. Wang, J. Tang, H. Liu, and B. Li, "Unsupervised sentiment analysis for social media images." in *IJCAI*, 2015, pp. 2378–2379.

[37] W. Lian, R. Henao, V. Rao, J. Lucas, and L. Carin, "A multitask point process predictive model," in *ICML*, 2015, pp. 2030–2038.