SIDE: Representation Learning in Signed Directed Networks

Junghwan Kim, Haekyu Park, Ji-Eun Lee, and U Kang
Seoul National University
Seoul, Korea
kjh900809,hkpark627,dreamhunter,ukang@snu.ac.kr

ABSTRACT

Given a signed directed network, how can we learn node representations which fully encode structural information of the network including sign and direction of edges? Node representation learning or network embedding learns a mapping of each node to a vector. The mapping encodes structural information on network, providing low-dimensional dense node features for general machine learning and data mining frameworks. Since many social networks allow trust (friend) and distrust (enemy) relationships described by signed and directed edges, generalizing network embedding method to learn from sign and direction information in networks is crucial. In addition, social theories are critical tool in signed network analysis. However, none of the existing methods supports all of the desired properties: considering sign, direction, and social theoretical interpretation.

In this paper, we propose SIDE, a general network embedding method that represents both sign and direction of edges in the embedding space. SIDE carefully formulates and optimizes likelihood over both direct and indirect signed connections. We provide socio-psychological interpretation for each component of likelihood function. We prove linear scalability of our algorithm and propose additional optimization techniques to reduce the training time and improve accuracy. Through extensive experiments on real-world signed directed networks, we show that SIDE effectively encodes structural information into the learned embedding.

CCS CONCEPTS

•**Information systems** → *Data mining;* •**Networks** → **Online social networks;**

KEYWORDS

Network Embedding, Representation Learning, Signed Directed Network

1 INTRODUCTION

Given a network with signed directed edges, how can we learn a vector representation of each node, encoding rich information on the network topology? Network embedding is one of the fundamental problems in network analysis and has received much interest from the data mining community recently [1, 2, 4, 7, 9, 21, 23, 28–31, 33, 34, 37]. Network embedding maps nodes into low-dimensional vector space that summarizes various aspects of network topology and link structure. Consequently, the mapped vectors provide features for conventional machine learning and data mining frameworks to solve various tasks, involving node classification [25], link prediction [16] and clustering [32]. In addition, the proximity structure in embedding space provides alternative to random walk with restart methods [11, 12, 22, 26, 35, 36] for ranking nodes.

Many social networks, such as Epinions[1] and Slashdot[2], allow users to form positive (trust, friendship) or negative (distrust, opposition) connections to other users. Negative links contain additional information [13] that boosts the performance in multiple tasks such as link sign prediction [10] and node classification [27] in signed networks. In addition, link directions are significant predictors of future link formation [6]. Most existing network embedding methods, however, only focus on modeling basic symmetric link structure, failing to exploit additional useful information in negative links and link directions. Modeling sign and direction in network embedding framework introduces the following challenges: consistent interpretation of both signs, representation of individual linking tendency, and utilization of multi-step connections.

There are several previous works on network embedding methods for signed networks. Spectral approaches [14, 38] cannot represent asymmetry in direction due to constraints that spectral theory imposes. Wang et al. [31] adopt neural network architecture to separate positive and negative connections, but their work also does not apply to directed network. Yuan et al. [37] exploit log-bilinear model which exploits blackbox model of embedding space and thus does not provide social theoretical interpretation.

In this paper, we propose SIDE (SIgned Directed network Embedding), a general network embedding method for signed directed networks. SIDE successfully represents proximity in signed directed network as a compact low-dimensional vector. For example, as presented in Figure 1, SIDE separates clusters more clearly in the embedding space than other baseline methods. In the embedding space, the friendly (blue lines) and unfriendly (red lines) relationships among social entities are effectively represented as distances between entities in the embedding space. Table 1 compares SIDE with other algorithms in various perspectives; SIDE is the only

---

[1] www.epinions.com
[2] slashdot.org
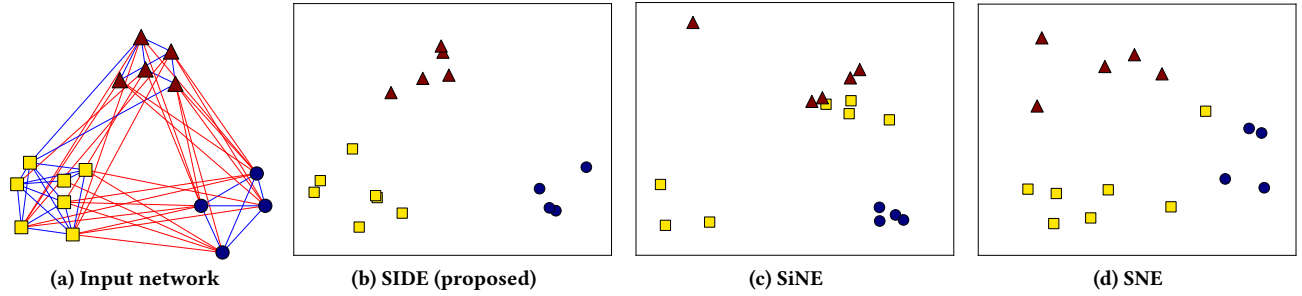[3] http://konect.uni-koblenz.de/networks/ucidata-gama

Figure 1: Visualization of the Eastern Central Highlands of New Guinea network[3][24]. Figure 1a represents the input network with favorable (blue lines) and hostile (red lines) relations. Figures 1b, 1c, and 1d show the embedding result of SIDE, SiNE, and SNE, respectively. SIDE best preserves the group structure marked by different shapes of nodes.

Table 1: Comparison of SIDE and other embedding algorithms. Our proposed method SIDE applies to the most general settings and shows the best performance for all metrics.

Method	Consider sign?	Consider direction?	Predictive accuracy	Speed
N2V [7]	No	Yes	Low	Fast
MF [8]	Yes	Yes	Medium	Medium
BNS [38]	Yes	No	Medium	Slow
SiNE [31]	Yes	No	Medium	Slow
SNE [37]	Yes	Yes	Low	Medium
SIDE	**Yes**	**Yes**	**High**	**Fast**

method which satisfies the desired properties: 1) considering sign and direction, 2) accurate, and 3) fast.

We base our network embedding method on truncated random walk and devise a general likelihood formulation for signed directed connections that represent both positive and negative edges consistently. Bias factors are employed in the likelihood function to model individual connectivity. We also generalize random walk sampling process and likelihood formulation to model multi-step relationship including both sign and direction. SIDE closely associates vector space geometry with social phenomena in networks such as homophily, preferential attachment, and balance theoretic behaviours. The association decomposes dual factors of link formation and provides a solid basis to apply our method in general analysis of signed directed networks.

Our contributions are listed as follows:

- **Algorithm.** We propose SIDE, a novel network embedding method on signed directed network (Section 3). By leveraging both sign and direction in the network, it achieves the state-of-the-art accuracy in link sign prediction task.
- **Analysis.** We perform detailed analysis of SIDE. We analyze and show that components in our formulation are associated with the socio-psychological interpretation (Section 3.4 and 4.3). In addition, complexity analysis proves that our algorithm shows scalability, linear in the number of nodes (Section 3.6).
- **Performance.** SIDE shows outstanding performance in terms of both accuracy and speed (Section 4). SIDE achieves the state-of-the-art performance in link sign prediction task. In addition, the learning process is up to 1.8× faster than other embedding methods with neural architecture.

Table 2: Table of symbols.

Symbol	Definition
$G = (V, E)$	input network
s	sign assignment function
$\mathcal{D}$	the set of co-occurring node pairs
u, v	nodes
v'_j	a noise node
W^{out}	out embedding matrix
W^{in}	in embedding matrix
$b^{in,+}, b^{in,-}$	positve/negative in-bias vectors
$b^{out,+}, b^{out,-}$	positive/negative out-bias vectors
σ	sigmoid function
w	the number of walks per node
l	the number of steps per walk
k	the context window size
n	the number of noise sampling
d	the embedding dimension
λ	the regularization parameter

The code of our method and datasets used in this paper are available at http://datalab.snu.ac.kr/side. The rest of this paper is organized as follows. We describe preliminaries in Section 2. Our proposed model SIDE is presented with detailed analysis in Section 3, followed by experimental results in Section 4. After discussing related works in Section 5, we conclude in Section 6.

2 PRELIMINARIES

In this section, we first describe socio-psychological theories which provide better understanding of the link formation process and the link sign structure (Section 2.1). Then we present random walk based network embedding formulation (Section 2.2). Table 2 lists the symbols used in this paper.

2.1 Socio-Psychological Theories

We employ socio-psychological theories to understand the formation of signed directed links. We first identify homophily and preferential attachment as two driving forces of link formation. Then we describe balance theory to deduce sign of multi-step connection.

Link Formation. We decompose causes of link formation into two factors: *similarity* and *connectivity*. Homophily [17] explains similarity factor in link formation. Homophily is portrayed as "Birds

of a feather flock together", which indicates that a node is more likely to connect with nodes that have similar properties. Similarity is naturally symmetric and transitive. Therefore, homophily is well modeled with distance in metric space by putting similar nodes closely together and dissimilar nodes far apart from each other.

Connectivity factor is characterized by preferential attachment [20]. It states that nodes with higher connectivity are more likely to form additional links. According to preferential attachment, the link formation likelihood is asymmetric and determined by individual connectivity of nodes. For example, following celebrity in real world directed networks is not likely to be reciprocated. In signed directed network, positive/negative and in/out links determine four different types of connectivity. For example, celebrities in a social network tend to receive more in-links than out-links due to their popularity, while advertisers tend to form many out-links in order to disseminate information. Because of the asymmetric and personalized nature, symmetric metric space is not adequate to model preferential attachment.

Link sign structure. Balance theory [3] is a well-established socio-psychological theory that states the following four rules: "A friend of my friend is my friend," "A friend of my enemy is my enemy," "An enemy of my friend is my enemy," and "An enemy of my enemy is my friend." The theory permits only even number of negative edges in a triad of signed network.

Balance theory provides basic rules to infer signs of multi-step relations. Given a path from a node u to another node v, adding hypothetical edges from u to every node on the path forms multiple triads. By sequentially applying balance theory on these triads, eventually the sign of the relationship between u and v can be inferred. As a result, the sign of multi-step connection is a successive multiplication of signs on edges along the path.

2.2 Network Embedding Formulation

Network embedding method learns a mapping of each node to a vector and encodes a link structure of network into a proximity structure in vector space. By preserving the information on network topology, the embedding provides informative features for off-the-shelf machine learning algorithms to solve tasks on networks such as node classification, link prediction and community detection.

The network embedding problem is formulated as follows. Let $G = (V, E)$ be a given network with nodes V and edges E. The network embedding method aims to learn a function $f : V \rightarrow \mathbb{R}^d$ which maps each node $v \in V$ to a d-dimensional vector. The embedding function is parameterized by a $|V| \times d$ embedding matrix W which consists of the d-dimensional row vector for each node.

Random walk based network embedding [23] learns two embedding functions corresponding to "target" and "neighborhood", respectively. The embeddings encode network structure by exploiting a language model. In the rest of this section, we describe the method in two stages: random walk generation and likelihood optimization. By generating multiple truncated random walks, the method transforms graph structure into sequential structure. Then the likelihood adopted from the language model is optimized to learn proximity structure of the graph from the node co-occurrence frequencies in the random walks.

Random Walk Generation. In the first stage, the method generates multiple truncated random walks on the graph. The result is node sequences which reveal proximity among nodes. Each step of walks chooses next node according to transition probabilities proportional to weights on edges. Then we generate multiple co-occurring pairs for calculation and optimization of the likelihood. Two nodes are defined to be a co-occurring pair if two nodes are placed within short distance in random walk sequences. If node pairs are connected with more heavily weighted and shorter paths, they are more likely to be within fewer steps from each other in the random walk node sequences. Consequently, the co-occurrence statistic of node pairs encodes proximity between nodes.

The truncated random walk is efficient in generating sample pairs of proximate nodes [7]. While each step of walk requires only one random number generation for choosing the next node, the chosen node forms pairs with multiple proximate nodes increasing the effective sampling rate. Consequently, better time complexity is achieved by leveraging increased effective sampling rate.

Likelihood Optimization. In the second stage, the method learns vector representation according to a neural language model, especially skipgram with negative sampling (SGNS) [18, 19]. SGNS is formulated as maximum likelihood on co-occurrence of node pairs. Direct prediction on neighboring nodes from target node using softmax function requires infeasible amount of parameter updates for each node pair. In order to limit the number of parameters updated in each step, SGNS employs negative sampling approach which defines binary classification on co-occurrence of node pairs rather than the direct prediction of co-occurring node. Therefore, likelihood model of SGNS predicts whether a pair of nodes co-occurs or not in the simulated random walk. The likelihood of u linking to v is formulated as follows:

$$P(u, v) = \sigma(W_u \cdot W_v') = \frac{1}{1 + \exp(-W_u \cdot W_v')} \tag{1}$$

where σ is a sigmoid function. W_u and W_v' are the "target" and "neighborhood" embedding vectors, respectively. The likelihood performs binary classification on whether node pairs co-occur or not. Intuitively, a pair of nodes with larger inner product value $W_u \cdot W_v'$ has a higher likelihood of co-occurrence.

The negative log-likelihood objective is defined as follows:

$$\sum_{(u,v) \in \mathcal{D}} [-\log P(u, v) + \sum_{j=1}^{n} -\log(1 - P(u, v_j'))] \tag{2}$$

where $\mathcal{D}$ is a set of co-occurring node pairs in random walk corpus and v_j' is a randomly sampled noise node. Each co-occurring example pair $(u, v) \in \mathcal{D}$, sampled from the first stage, is used to take one step of gradient descent to optimize the first part of the Equation (2). For each co-occurring example (u, v), multiple noise pairs are defined as (u, v_j') where v_j' are randomly sampled nodes. The noise pairs constitute negative examples for the binary classification and are used in the second part of Equation (2). Gradient descent update for noise samples push two randomly sampled nodes apart from each other. This prevents all nodes from converging to a single point and penalizes unconnected node pairs of being close to each other. Multiple noise pairs are sampled for each co-occurring node pair to account for the imbalance of positive and noise pairs due to the link sparsity in real world network.

Junghwan Kim, Haekyu Park, Ji-Eun Lee, and U Kang

3 PROPOSED METHOD

In this section, we describe SIDE, our proposed method for network embedding on signed directed network.

3.1 Overview

Let G be a given signed directed network with nodes V, edges E and sign assignment function $s : E \to \{-1, +1\}$. We aim to learn target and neighborhood embedding functions $f, g : V \to \mathbb{R}^d$ which depict link structure including sign and direction. We exploit the random walk based network embedding framework described in Section 2.2; embedding function is derived by maximizing likelihood over node pairs sampled from directed random walk. Embedding signed directed network entails the following challenges:

(1) **Multi-step relationship.** How can we generalize single-step sign and direction to multi-step neighbors?
(2) **Negative edges.** How can we encode negative links in embedding space without hindering positive proximity?
(3) **Individual connectivity.** How can we separately model connectivity factor from similarity factor in link formation?

We propose the following main ideas to address the challenges.

(1) **Sign and direction aggregation** generalizes the notion of single-step sign and direction to multi-step connections. Signs along the path are aggregated according to balance theory and direction follows topological order of the graph (Section 3.2).
(2) **Signed proximity term** assigns high likelihood for proximate positively connected pairs and distant negatively connected pairs. Positive pull and negative push are balanced under the maximum likelihood framework (Section 3.3).
(3) **Bias terms** model individual node connectivity separately from inner product similarity. We distinguish positive/negative and in/out biases to incorporate asymmetry in connecting pattern of each node (Section 3.3).

Algorithm 1 summarizes the procedure of SIDE and shows how the main ideas are implemented. SIDE first simulates multiple random walks (lines 3 to 7) and generates co-occurring pairs (u, v) of node from the simulated walks (line 9). Then, for each pair (u, v), gradient descent steps for both sampled pairs (u, v) (line 10) and noise pairs (u, v') (line 13) are performed to update embedding vectors and biases. In the following sections, we describe our framework details, present social theoretical analysis, introduce additional optimization techniques, and perform complexity analysis of SIDE.

3.2 Sign and Direction Aggregation

In the first stage of SIDE, the pairs of nodes are sampled from a random walk process. The sampling process first generates truncated random walks starting from each seed node. Each step of the walk follows directed edges until the required length is satisfied; the order of nodes in the random walk conforms to the topological order in the directed network. If the random walk encounters dead end, the remaining steps restart from the seed node. The resulting walk sequence consists of visited nodes and signs on the followed edges. Then co-occurring pairs of nodes within a window of size k are selected from the walk sequences. We define sign and direction for each co-occurring pair aggregating information along the path.

Algorithm 1 SIDE algorithm.

Input: signed directed network $G = (V, E)$ with s,
    dimension d, walks per node w, steps per walk l,
    context size k, noise sampling size n,
    regularization parameter λ

Output: out and in embedding matrix $W^{out}, W^{in} \in \mathbb{R}^{|V| \times d}$,
    positive in-link bias $b^{in,+}$, negative in-link bias $b^{in,-}$,
    positive out-link bias $b^{out,+}$, negative out-link bias $b^{out,-}$
    // Initialization

1: Initialize W^{out} and W^{in} to random values and $b^{in,+}$, $b^{in,-}$,
    $b^{out,+}$, and $b^{out,-}$ to zeros
2: $Walks = \{\}$
    // Random Walk Generation
3: **for** $i = 1$ **to** w **do**
4:     **for all** $v \in V$ **do**
5:         Generate a random walk of length l starting from v and
        append to $Walks$
6:     **end for**
7: **end for**
    // Likelihood Optimization
8: **for all** $walk \in Walks$ **do**
9:     **for all** (u, v) within distance k in $walk$ **do**
10:         Update W_u^{out}, W_v^{in}, $b_u^{out, sign(u,v)}$, and $b_v^{in, sign(u,v)}$ by
        taking gradient descent step according to Equation (5)
11:         **for** $j = 1$ **to** n **do**
12:             Randomly sample $v' \in V$
13:             Update W_u^{out} and $W_{v'}^{in}$ by taking gradient descent step
            according to Equation (6)
14:         **end for**
15:     **end for**
16: **end for**

Sign of the node pair is inferred according to the balance theory as described in Section 2.1: sign on multi-step connection is multiplication of edge signs along the path. The sign is negative if there are odd number of negative edges along the path while it is positive if there are even number of negative edges.

We define directed node pairs (u, v) if u precedes v from a sequential order in a random walk. This means that there is a directed path from u to v in the network since a sequential order in random walk conforms to a topological order in the network. By identifying and parameterizing source and destination nodes separately, it is possible to model asymmetric direction.

3.3 Likelihood Formulation

We now describe maximum likelihood formulation that links embedding vectors to likelihood of sampled pairs. The objective function is defined as follows:

$$J = \sum_{(u,v) \in \mathcal{D}} [-\log P(u, v) + \sum_{j=1}^{n} -\log P(u, v_j')]$$

$$+ \frac{\lambda}{2}(\|b^{in,+}\|^2 + \|b^{in,-}\|^2 + \|b^{out,+}\|^2 + \|b^{out,-}\|^2) \quad (3)$$

where $(u, v) \in \mathcal{D}$ is a node pair with aggregated sign and direction as defined in the Section 3.2. For each pair (u, v), n noise samples

v_j' are randomly selected to form noise pairs. The latter part of the objective function regularizes bias terms in the likelihood function.

The likelihood $P(u, v)$ is defined as follows:

$$P(u, v) = \begin{cases} \sigma(W_u^{out} \cdot W_v^{in} + b_u^{out,+} + b_v^{in,+}) & \textbf{if } sign(u, v) > 0 \\ \sigma(-W_u^{out} \cdot W_v^{in} + b_u^{out,-} + b_v^{in,-}) & \textbf{if } sign(u, v) < 0 \\ \sigma(-W_u^{out} \cdot W_v^{in}) & \textbf{if } v \text{ is a noise} \end{cases}$$

$$(4)$$

where $sign(u, v)$ represents the aggregate sign defined in Section 3.2. The first, second, and third equations define likelihood for positive, negative, and noise pairs, respectively. The Equations (4) consist of two components: signed proximity term $\pm W_u^{out} \cdot W_v^{in}$ and bias terms $b_u^{out,\pm}, b_v^{in,\pm}$.

Signed proximity term. The first component of the likelihood function is signed inner product proximity term $\pm W_u^{out} \cdot W_v^{in}$. For positively connected nodes, likelihood value increases as the inner product term increases. Negative and noise pairs, on the other hand, have higher likelihood value when the inner product similarity is lower. By maximizing the objective function, embedding vectors are learned to assign high similarity values for positively connected nodes while low similarity values for negatively connected nodes. The balance between positive pull and negative push is achieved within the maximum likelihood framework. As a result, nodes with similar signs on in- and out- links learn similar in- and out-embedding vectors, respectively. On the other hand, nodes with oppositely signed neighborhood structure are placed far apart from each other in the embedding vector space.

Bias terms. We employ bias terms $b_u^{out,\pm}, b_v^{in,\pm}$ as a second component of the likelihood function. According to preferential attachment, larger connectivity induces higher likelihood of additional link formation. Our likelihood formulation models this connectivity with bias terms. High bias term values result in high likelihood of link formation even if signed proximity term is small. The bias terms depict asymmetric and personalized nature of node connectivity. Bias terms are separately defined for each node and each type of role in link formation.

As shown in Equation (4), likelihood function of each link contains two bias terms: out-link bias of a source node and in-link bias of a destination node. Source and destination are determined by the aggregated direction as described in Section 3.2. As a result, it is possible to assign asymmetric likelihood values on two reciprocal edges that link the same node pair. In addition, the likelihood of each link depends on which nodes participate in the link and formulates personalized nature of the link formation. Sign and direction define four types of role that a node can participate in link formation. For each node $u \in V$, we define four distinct bias factors corresponding to each type of role: positive in-link bias $b_u^{in,+}$, negative in-link bias $b_u^{in,-}$, positive out-link bias $b_u^{out,+}$, and negative out-link bias $b_u^{out,-}$. We denote $|V| \times 1$ bias vectors corresponding to each type as $b^{in,+}, b^{in,-}, b^{out,+}$, and $b^{out,-}$.

Gradient descent optimization. We train our model using gradient descent optimization. The derivative required to update each parameter one step in gradient descent stage is distinct for co-occurring pair and noise pair. For co-occurring pair (u, v), two weight vectors W_u^{out}, W_v^{in} and two bias factors $b_u^{out,sign(u,v)}$, $b_v^{in,sign(u,v)}$ are updated. The derivative of objective function

$J_{(u,v)} = -\log P(u, v) + \frac{\lambda}{2}(|b_u^{out,sign(u,v)}|^2 + |b_v^{in,sign(u,v)}|^2)$ corresponding to the weight vectors and the bias factors of co-occurring pair (u, v) is as follows:

$$\frac{\partial J_{(u,v)}}{\partial W_u^{out}} = -sign(u, v) W_v^{in}(1 - P(u, v))$$

$$\frac{\partial J_{(u,v)}}{\partial W_v^{in}} = -sign(u, v) W_u^{out}(1 - P(u, v))$$

$$\frac{\partial J_{(u,v)}}{\partial b_u^{out,sign(u,v)}} = -(1 - P(u, v)) + \lambda b_u^{out,sign(u,v)}$$

$$\frac{\partial J_{(u,v)}}{\partial b_v^{in,sign(u,v)}} = -(1 - P(u, v)) + \lambda b_v^{in,sign(u,v)}$$

$$(5)$$

where the likelihood $P(u, v)$ is determined according to $sign(u, v)$ as in the Equation (4). The regularization on bias terms is essential in order to prevent bias values from diverging. If regularization of bias terms were not applied, the gradient updates for bias are always positive and keep increasing the value of the bias.

For noise pair (u, v_j'), only two weight vectors $W_u^{out}, W_{v_j'}^{in}$ are updated. The derivative of objective function $J_{(u, v_j')} = -\log P(u, v_j')$ corresponding to noise pair (u, v_j') is as follows:

$$\frac{\partial J_{(u, v_j')}}{\partial W_u^{out}} = W_{v_j'}^{in}(1 - P(u, v_j'))$$

$$\frac{\partial J_{(u, v_j')}}{\partial W_{v_j'}^{in}} = W_u^{out}(1 - P(u, v_j'))$$

$$(6)$$

where the likelihood $P(\cdot, \cdot)$ is defined as the third equation in Equation (4).

3.4 Social Theoretical Analysis

In this section, we describe social theoretical interpretation of our model. Firstly, we claim that signed proximity term is consistent with balance theory. Then, we identify the role of bias terms in preferential attachment. These two components in the likelihood formulation establish two factors determining link formation as discussed in Section 2.1: homophily and preferential attachment.

According to balance theory, positively connected nodes tend to have similar sign on their shared neighbors and have similar neighborhood structure. Through signed proximity term, our model places nodes with similarly signed neighborhood closely together and nodes with oppositely signed neighborhood far apart. We deduce that positively connected nodes have closely placed embedding vectors while negatively connected nodes are placed far apart in embedding space. Balance theory is consistent with this embedding distance structure. For example, a triad with all positive edges constructs three nodes closely placed together, a triad with two negative edges constructs one node far apart from the other two closely placed nodes, and a triad with all negative edges places all three nodes far apart from each other. The triad not allowed in balance theory is incompatible with the transitivity of closeness in the embedding space. Therefore, our embedding method naturally encourages the embedding vectors to be learned to follow balance theory.

The bias terms model individual connectivity, while inner product term models symmetric similarity between nodes. In the learning process, every pair example increases bias values of corresponding type. We interpret bias terms of each node as a node connectivity; it is expected that large out-link bias and large in-link bias are learned for nodes with high out-degree and nodes with high in-degree, respectively. According to the likelihood formulation in (4), link formation likelihood increases as the bias term increases. Therefore, the bias terms model preferential attachment process where bias terms for nodes with high connectivity are inclined toward increasing link formation likelihood.

3.5 Additional Optimization Techniques

We suggest two additional strategies to accelerate the learning process of networks: deletion of nodes with degree one and subsampling of high-degree nodes.

A large portion of nodes in real-world networks have degree one, since degree distributions in real-world networks follow a power-law distribution. For example, more than 20% of nodes in our dataset have degree one as shown in Table 3. In the random walk sequences, a node with positive degree one appears with the only node linked to it. Consequently the degree-one node shares all of its neighborhood structure with its only neighbor; therefore the learning process performs huge number of redundant calculations. To prevent this, we eliminate nodes with positive degree-one and learn embedding for reduced network. Then, for each eliminated positive degree-one node, embedding vector is copied from the embedding vector of the only neighbor. By eliminating nodes with degree one, we reduce the number of parameters to learn in the optimization process.

Real-world networks also contain nodes with very high degrees according to the power-law distribution. High-degree nodes are connected to a huge number of nodes, but most of the connections are uninformative at inferring proximity structure of either high-degree nodes themselves or neighboring nodes. Inspired by subsampling in SGNS model [19], we subsample high-degree nodes in the process of random walk generation. Each node u is thrown away with probability $1 - \sqrt{\frac{t}{p(u)}}$ where $p(u)$ is the degree of u divided by the number of edges and t is a threshold hyperparameter for $p(u)$. We fix t to 0.001 which is a frequent default value in the SGNS model. Subsampling enables more informative random walk sampling process because of the increased effective window size.

3.6 Time Complexity Analysis

In this section, we provide proofs on linear scalability of the time complexity of SIDE. Theorem 3.4, which is the main result of this section, is proved using three lemmas for each stage of SIDE. We assume that the graph is stored in memory so that each step in random walk requires constant time.

LEMMA 3.1. *(Time Complexity of Random Walk Generation) Random walk generation takes $O(|V|wl)$ time.*

PROOF. We generate l steps in w random walks for $|V|$ nodes. Each random walk step requires random choice of out-linked nodes and takes constant time. Combining the above two results, the time complexity of random walk generation process is $O(|V|wl)$. □

Table 3: Statistic of the Datasets.

	[a]Epinions	[b]Slashdot	[c]Wikipedia
# of nodes	131,828	82,140	7,118
# of edges	841,372	549,202	103,675
% of positive edges	85.3 %	76.1 %	78.4 %
% of negative edges	14.7 %	23.9 %	21.6 %
% of nodes with positive degree one	38.2 %	25.5 %	24.9 %

[a]http://www.trustlet.org/wiki/Extended_Epinions_dataset
[b]http://dai-labor.de/IRML/datasets
[c]http://snap.stanford.edu/data/wiki-Vote.html

LEMMA 3.2. *(Time Complexity of Pair Sampling Process) Pair sampling takes $O(|V|wlk^2)$ time.*

PROOF. We generate k node pairs for each random walk step, which sums up to $|V|wlk$. Each node pair sampling requires the calculation of signs which takes $O(k)$ time in average, because we need to count the number of positive and negative edges in the path between the node pairs. Therefore, the time complexity for pair sampling is $O(|V|wlk^2)$. □

LEMMA 3.3. *(Time Complexity of Gradient Descent Learning) Gradient descent learning takes $O(|V|wlk(n + 1)d)$ time.*

PROOF. We perform n noise sampling for $|V|wlk$ connected pairs. The number of node pairs sampled from random walk is $|V|wlk$ and the number of node pairs from noise sampling is $|V|wlkn$, summing up to $|V|wlk(n + 1)$. Since a gradient descent update for each node pair takes $O(d)$, the total time complexity of gradient descent learning is $O(|V|wlk(n + 1)d)$. □

THEOREM 3.4. *(Time Complexity of SIDE) The training of SIDE takes $O(|V|)$ time.*

PROOF. SIDE consists of three stages: random walk generation, pair sampling, and gradient descent learning. Combining the preceding three Lemmas, and considering that the parameters w, l, k, n, and d are constants, SIDE shows the time complexity linear in the number of nodes $|V|$. □

4 EXPERIMENTS

In this section, we experimentally evaluate the performance of SIDE to answer the following questions.

- **Q1. (Sign Prediction)** How predictive is SIDE at inferring unobserved link signs? (Section 4.2)
- **Q2. (Representation)** How effectively does SIDE represent signed directed relationships? (Section 4.3)
- **Q3. (Speed and Scalability)** How fast and scalable is the training of SIDE compared to the baselines? (Section 4.4)
- **Q4. (Optimization)** How effective are optimization techniques of SIDE in terms of time and accuracy? (Section 4.5)

4.1 Experimental Setup

Data. We conduct experiments on three real-world datasets of signed directed social networks: Epinions, Slashdot, and Wikipedia. Epinions is a product review site where users can form trust and distrust relationships with other users. Slashdot is a technology news site which allows users to annotate other users as friends or foes. Wikipedia is an encyclopedia site where users vote for or

Table 4: Performance on prediction of link sign. SIDE shows the state-of-the-art performance in link sign prediction task.

		SIDE (proposed)	FE	N2V	MF	BNS	SNE	SiNE
	Epinions	**0.967**	0.951	0.764	0.920	0.893	0.820	0.860
AUC	Slashdot	**0.889**	0.889	0.697	0.877	0.842	0.746	0.816
	Wiki	**0.901**	0.879	0.648	0.875	0.861	0.762	0.790
	Epinions	**0.972**	0.960	0.893	0.957	0.948	0.924	0.922
F1	Slashdot	**0.911**	0.906	0.811	0.910	0.895	0.874	0.887
	Wiki	**0.918**	0.907	0.879	0.913	0.901	0.882	0.882

against other users in order to determine admin promotion. The statistics of the datasets are summarized in Table 3.

Baselines. We compare SIDE to a feature engineering method and five embedding methods. The competitors are as follows:

- Feature Engineering (FE) [15]. This model defines two types of hand-engineered features for each edge: signed/directed degrees of two participating nodes and sign/direction configurations of two-step paths from the source to the target.
- Node2vec (N2V) [7]. This is a random walk based network embedding method for unsigned network. Embedding is learned on positive subgraph with this method.
- Matrix Factorization (MF) [8]. This method learns low rank structure of signed network by matrix factorization.
- Balanced Normalized Signed Laplacian (BNS) [38]. This method applies a spectral embedding on modified Laplacian matrix which balances positive and negative links.
- Signed Network Embedding (SNE) [37]. This method utilizes log-bilinear model with random walk sampling.
- Signed Network Embedding (SiNE) [31]. This model trains deep neural network to make a distinction between positively connected nodes and negatively connected nodes.

Parameter. For random walk generation and embedding dimension, we use the parameter settings in [23]: $w = 80$, $l = 40$, and $d = 128$. We set optimal context size k, noise sampling size n, and regularization parameter λ differently to each dataset to gain the best performance. For baseline methods, we set the dimension to 128 and use the same settings for other parameters as those suggested in their papers, respectively.

4.2 Link Sign Prediction

We assess the predictive performance of SIDE in link sign prediction task. Link sign prediction is the task of predicting unobserved signs of existing edges in the test set. In order to focus on the performance of embedding methods, we train a simple logistic regression model with embedding vectors as features. Since an embedding vector is defined for each node, we derive an edge feature by combining two vectors of source and destination nodes. We use two methods for the combination: element-wise product and concatenation. Concatenation always outperforms element-wise product except for the results in Epinions dataset. We use 5-fold cross validation. Each training set is used to train both embedding vectors and logistic regression model.

Table 5: Analysis of Signed proximity terms. Signed proximity average values for positive and negative links are clearly separated, more than standard deviation.

	Positive		Negative		P-value	
	avg.	std.	avg.	std.	WT	KS
Epinions	3.685	1.827	-2.763	1.631	<0.0001	<0.0001
Slashdot	3.060	1.661	-0.745	1.408	<0.0001	<0.0001
Wikipedia	2.538	1.073	-0.392	1.573	<0.0001	<0.0001
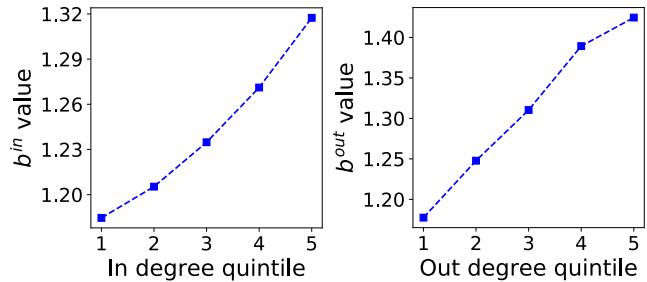
Figure 2: Correlation of bias and degree. Bias value increases as the degree quintile increases. This is consistent with the preferential attachment interpretation of bias terms.

We report AUC and $F1-score$ of methods as shown in Table 4. The higher the metrics are, the higher the predictive accuracy is. We make the following observations:

- SIDE achieves the best accuracy at link sign prediction in terms of both AUC and $F1-score$. This proves the superior predictive accuracy of our method.
- Feature engineering (FE) method is the only comparable competitor. FE uses features specifically targeted for link sign prediction task and outperforms all the generic embedding methods except for SIDE.
- SIDE and FE are the only methods that are based on socio-psychological theories. This shows the importance of social theory in explaining sign structure in the real-world networks.
- Node2vec (N2V) method, which is learned on positive subgraph, shows the worst accuracy because no sign information is fed in the learning process. This shows the importance of leveraging sign information to accurately infer unobserved edge sign.

4.3 Embedding Analysis

We examine the correctness of our algorithm by showing that learned parameter values are consistent with our design goals. We inspect two components in our likelihood formulation: signed proximity term and bias terms.

We design signed proximity term to distinguish distances of positive connections from those of negative connections. To check whether the inner product similarity term successfully represents link sign structure, we perform Welch's t-test and Kolmogorov-Smirnov test on inner product values between sets of positive and negative links. Welch's t-test examines whether two populations have equal means without any assumption on variance. Kolmogorov-Smirnov test is a test on the equality of two probability distribution. Smaller magnitude of p-values for both tests
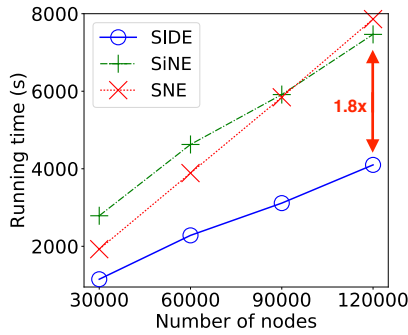
Figure 3: Scalability of SIDE. Our algorithm runs up to 1.8x faster than baseline methods, and shows linear scalability on the number of nodes with the smallest slope.

indicates a more distinct distribution of inner product similarity between positive edges and negative edges. Table 5 shows the result of the tests with average and standard deviation of positive and negative links in our dataset. P-values of the tests demonstrate the strong evidence on difference of both means and distributions between two edge sets. In addition, the averages for positive and negative link sets are clearly separated, more than the magnitude of corresponding standard deviations.

We verify the preferential attachment interpretation of the learning process of our bias terms. We divide nodes into 5 buckets according to degree quintiles. Then we calculate the average sum of positive and negative bias values for each bucket. We add positive and negative bias because indirect connections might increase bias of opposite signs. Figure 2 shows the relationship between average bias values and node degree. As node degree increases, the expected value of bias term increases in both in-degree and out-degree cases. This shows the validity of our model assumption that high degree nodes would learn high bias values. We only show the result for Epinions dataset since other datasets show similar pattern with different scale of bias values.

4.4 Speed and Scalability

To show the computational efficiency of SIDE, we compare SIDE with two recently proposed signed network embedding methods: SNE and SiNE. We extract principal submatrices of the Epinions network varying the number of nodes from 30,000 to 120,000 and estimate the training time of each method. In this experiment, we set both context size k and noise sampling size n to 5 which is optimal for Epinions dataset. SiNE is trained with only one epoch instead of 100 epochs as suggested in [31] because even one epoch takes much larger time than SIDE.

As shown in Figure 3, the running time of SIDE linearly increases with regard to the number of nodes, as described in Theorem 3.4. Also, SIDE runs up to 1.8× faster than other methods with smaller slopes. The superior speed of SIDE comes from the concise likelihood formulation and the limited number of parameters for each gradient descent update. In contrast, SiNE updates additional parameters of the deep neural network architecture and SNE updates embedding vectors for all nodes along the random walk sequence.
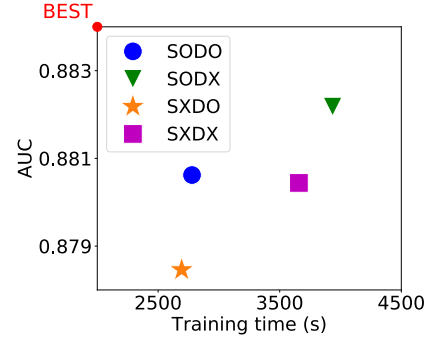

Figure 4: Running time-accuracy tradeoff of optimization techniques for SIDE. Subsampling increases accuracy while deletion of degree-one node decreases training time in the expense of small accuracy loss. Better running time-accuracy tradeoff is achieved by using both optimization techniques.

4.5 Effect of Optimization

We show the effects of optimization techniques, subsampling of high-degree nodes, and deletion of one-degree nodes, on the performance of SIDE. We set optimal context size k and noise sampling size n to 7 and 1, respectively. In Figure 4, S denotes subsampling of high-degree nodes and D denotes deletion of one-degree nodes. The activation of each optimization technique is indicated as O/X sign next to S and D. Note that the two techniques give a trade-off between the running time and the accuracy. The subsampling technique increases accuracy with longer running time. Subsampling discards high-degree node randomly; since the random walk continues until the required length is met, the sampling process takes a little longer. However, by skipping some uninformative nodes, sampled nodes provide more information to the algorithm. On the other hand, deletion of degree-one nodes decreases the accuracy with faster running time. The discarded degree-one node copies the embedding vector of the only node linked to it. Because of the smaller number of parameters to learn, the training time decreases at a cost of small loss in accuracy. Overall, employing both optimization techniques leads to increased accuracy and decreased training time as shown in Figure 4.

5 RELATED WORKS

Recently, there has been much interests on network embedding in data mining community [1, 2, 4, 7, 9, 21, 23, 28–30, 33, 34]. There are three general categories which existing methods fall into: models based on matrix factorization, models exploiting deep neural networks, and models learning from truncated random walks.

Matrix factorization is one of the most straightforward approach to reduce the adjacency matrix into low-dimensional space. Spectral embedding method [5] applies eigendecomposition on Laplacian matrix of the network in order to get useful low-dimensional representations. Cao et al. [1] suggested factorizing k-step transition matrix to learn global structural information. Mingdong et al. [21] introduced SVD on high-order proximity matrix to capture asymmetric transitivity in directed network.

Other models exploit the deep neural network to learn nonlinear structural information. Tian et al. [29] leveraged stacked

autoencoder to reconstruct normalized similarity matrix. Wang et al. [30] combined deep autoencoder and Laplacian eigenmap to model second-order and first-order proximities, respectively.

Random walk based methods utilize language model, especially skipgram model, to learn co-occurrence statistic from simulated random walks. Perozzi et al. [23] first suggested the framework of random walk based network embedding method. Grover and Leskovec [7] further extended the random walk process with distorted probability to mimic breadth first search (BFS) and depth first search (DFS). Chen et al. [4] proposed to additionally utilize group label and model group structure by group embedding.

However, none of the above approaches has straightforward generalization to learn network embedding for signed directed network. Some methods tackle signed network embedding in the context of spectral embedding [14, 38]. These methods, however, are applicable only to undirected networks and are not scalable because of the complexity of eigendecomposition. Wang et al. [31] exploit a deep learning framework for signed network. Although the framework leverages nonlinearity and complexity of neural network architectures, it does not model edge directions. Yuan et al. [37] extended log-bilinear model to support sign and direction modeling. Log-bilinear formulation does not consider balance theoretic nature of multi-step connections and shows suboptimal efficiency in training process. SIDE provides fast and linearly scalable method to learn network embedding in the most general settings where edges have both sign and direction. In addition, our embedding space is intuitively explainable in terms of socio-psychological theories.

6 CONCLUSIONS

We propose SIDE, a fast and accurate network embedding method to represent signed directed network. We construct SIDE to overcome three challenges: consistent interpretation of both signs, representation of individual linking tendency, and utilization of multi-step connections. SIDE interprets negative edges as an indication of remoteness, and models asymmetric direction with biases. Sign and direction aggregation along multi-step connections preserves sign and direction information. Social theoretical analysis shows the expressiveness and interpretability of our model components. We also propose techniques to optimize truncated random walk based network embedding frameworks. We prove that the complexity of our method is linear in the number of nodes. Experimental results show that SIDE outperforms competitors in link sign prediction task, well represents sign and direction into embedding space, and is learned efficiently.

ACKNOWLEDGMENT

REFERENCES

[1] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 891–900.

[2] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2016. Deep neural networks for learning graph representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 1145–1152.

[3] Dorwin Cartwright and Frank Harary. 1956. Structural balance: a generalization of Heider's theory. *Psychological review* 63, 5 (1956), 277.

[4] Jifan Chen, Qi Zhang, and Xuanjing Huang. 2016. Incorporate group information to enhance network embedding. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1901–1904.

[5] Fan RK Chung. 1997. *Spectral graph theory*. Vol. 92. American Mathematical Soc.

[6] Scott A Golder and Sarita Yardi. 2010. Structural predictors of tie formation in twitter: Transitivity and mutuality. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*. IEEE, 88–95.

[7] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

[8] Cho-Jui Hsieh, Kai-Yang Chiang, and Inderjit S Dhillon. 2012. Low rank modeling of signed networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 507–515.

[9] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label Informed Attributed Network Embedding. In *Proceedings of 10th ACM International Conference on Web Search and Data Mining (WSDM)*. ACM.

[10] Jinhong Jung, Woojeong Jin, Lee Sael, and U. Kang. 2016. Personalized Ranking in Signed Networks Using Signed Random Walk with Restart. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*. 973–978.

[11] Jinhong Jung, Namyong Park, Lee Sael, and U. Kang. 2017. BePI: Fast and Memory-Efficient Method for Billion-Scale Random Walk with Restart. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*. 789–804.

[12] Jinhong Jung, Kijung Shin, Lee Sael, and U. Kang. 2016. Random Walk with Restart on Large Graphs Using Block Elimination. *ACM Trans. Database Syst.* 41, 2 (2016), 12. DOI:http://dx.doi.org/10.1145/2901736

[13] Jérôme Kunegis, Julia Preusse, and Felix Schwagereit. 2013. What is the added value of negative links in online social networks?. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 727–736.

[14] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak. 2010. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 559–570.

[15] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World Wide Web*. ACM, 641–650.

[16] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology* 58, 7 (2007), 1019–1031.

[17] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

[20] Michael Mitzenmacher. 2004. A brief history of generative models for power law and lognormal distributions. *Internet mathematics* 1, 2 (2004), 226–251.

[21] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1105–1114.

[22] Haekyu Park, Jinhong Jung, and U. Kang. 2017. A comparative study of matrix factorization and random walk with restart in recommender systems. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*. 756–765.

[23] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. ACM, 701–710.

[24] Kenneth E Read. 1954. Cultures of the central highlands, New Guinea. *Southwestern Journal of Anthropology* 10, 1 (1954), 1–43.

[25] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93.

[26] Kijung Shin, Jinhong Jung, Lee Sael, and U Kang. 2015. BEAR: Block Elimination Approach for Random Walk with Restart on Large Graphs. In *SIGMOD*.

[27] Jiliang Tang, Charu Aggarwal, and Huan Liu. 2016. Node classification in signed social networks. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 54–62.

[28] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 1067–1077.

[29] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. 2014. Learning Deep Representations for Graph Clustering. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 1293–1299.

[30] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1225–1234.

[31] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. 2017. Signed Network Embedding in Social Media. In *Proceedings of the 2017 SIAM international conference on data mining*. SIAM.

[32] Scott White and Padhraic Smyth. 2005. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, 274–285.

[33] Cheng Yang and Zhiyuan Liu. 2015. Comprehend deepwalk as matrix factorization. *arXiv preprint arXiv:1501.00358* (2015).

[34] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *Proceedings of The 33rd International Conference on Machine Learning*. 40–48.

[35] Minji Yoon, Woojeong Jin, and U. Kang. 2018. Fast and Accurate Random Walk with Restart on Dynamic Graphs with Guarantees. In *The Web Conference, 2018*.

[36] Minji Yoon, Jinhong Jung, and U. Kang. 2018. TPA: Fast, Scalable, and Accurate Method for Approximate Random Walk with Restart on Billion Scale Graphs. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*.

[37] Shuhan Yuan, Xintao Wu, and Yang Xiang. 2017. SNE: Signed Network Embedding. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 183–195.

[38] Quan Zheng and David B Skillicorn. 2015. Spectral embedding of signed networks. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 55–63.