SIDE: Representation Learning in Signed Directed Networks

Anonymous
Anonymous
Anonymous

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 useful 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. However, existing network embedding methods either do not consider both sign and direction of edges, or show limited performance in terms of accuracy and scalability.

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. Additional optimization techniques reduce the training time and improve accuracy. We provide socio-psychological interpretation for each component of likelihood function and prove linear scalability of our algorithm. Through extensive experiments on real-world signed directed networks, we show that SIDE effectively encodes useful structural information into the learned embedding.

CCS CONCEPTS

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

KEYWORDS

Network Embedding, Feature Learning, Signed Directed Network

ACM Reference format:
Anonymous . 2018. SIDE: Representation Learning in Signed Directed Networks . In *Proceedings of ACM International Conference on Web Search and Data Mining, Los Angeles, California, USA, February 2018 (WSDM'18),* 9 pages.
DOI: 10.475/123_4

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, 8, 10, 19, 20, 24–27, 30–32]. Network embedding maps nodes into low-dimensional vector space that summarizes various aspects of network topology and link structure. Consequently, the mapped vectors provide useful features for conventional machine learning and data mining frameworks to solve various tasks, involving node classification [22], link prediction [14] and clustering [28].

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 [11] that boosts the performance in multiple tasks such as link sign prediction [13] and node classification [23] in signed networks. In addition, link directions are significant predictors of future link formation [7]. 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 asymmetry within symmetric metric space, and utilization of multi-step connections.

There are several previous works on network embedding methods for signed networks. Some works took spectral approaches on signed network embedding [12, 33]. However, these models cannot represent asymmetry in direction due to the constraints that spectral theory impose. Wang et al. [27] adopt neural network architecture to separate positive connections from negative connections, but their work does not apply to directed network and lacks global perspective in optimization. Yuan et al. [32] exploit random walk based network embedding framework with log-bilinear model which was proven to show inferior efficiency compared to skipgram with negative sampling approach in language model literatures [17]. In addition, all previous works do not associate vector space geometry with social phenomena in networks such as homophily, preferential attachment, and balance theory.

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

DOI: 10.475/123_4

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

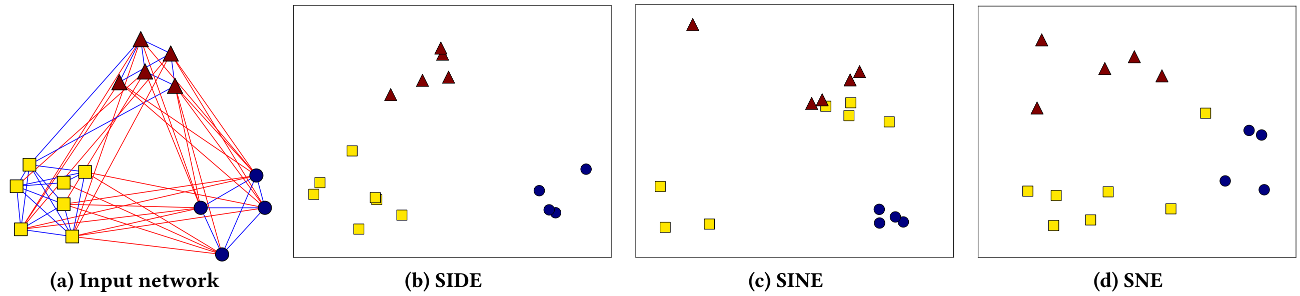

Figure 1: Visualization of the Eastern Central Highlands of New Guinea network[3] [21]. Figure 1a represents the input network with favorable (blue lines) and hostile (red lines) relations. Figures 1b, 1c, and 1d show the embedding result of SIDE, SiNE, and SNE, respectively. SIDE best preserves the 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 [8]	No	Yes	Low	Fast
MF [9]	Yes	Yes	Medium	Medium
BNS [33]	Yes	No	Medium	Slow
SiNE [27]	Yes	No	Medium	Slow
SNE [32]	Yes	Yes	Low	Medium
SIDE	**Yes**	**Yes**	**High**	**Fast**

We base our network embedding method on truncated random walk as in [4, 8, 20]. We 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 asymmetry in direction. We also generalize random walk sampling process and likelihood formulation to model multi-step relationship including both sign and direction. This 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 and general network embedding method on signed directed network (Section 3). By leveraging both sign and direction information in the network, the algorithm achieves the state-of-the-art accuracy in link sign prediction task.
- **Analysis.** We perform detailed analysis of our method. The components in our formulation are associated with the interpretation in terms of socio-psychological theories (Section 3.5). In addition, complexity analysis proves that our algorithm shows scalability, linear in the number of nodes (Section 3.6).
- **Performance.** SIDE shows favorable performance in terms of both accuracy and speed. 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.

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.

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	target nodes
v'_j	a noise node
W	target embedding matrix
W'	neighborhood 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

2 PRELIMINARIES

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

2.1 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 useful 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 $f : V \longrightarrow \mathbb{R}^d$ 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 [20] learns 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. 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, which counts how many times node pairs appear within a short distance, encodes proximity between nodes.

The truncated random walk is efficient in generating sample pairs of proximate nodes [8]. 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) [16, 17]. 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 function 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), noise example pairs are defined as u paired with randomly sampled nodes v'_j, and are used to take gradient descent steps optimizing the second part. Noise samples constitute negative examples for the binary classification. 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.

2.2 Socio-Psychological Theories

We employ socio-psychological theories to understand the factors that determine the formation of signed directed links. We first review homophily and preferential attachment to identify two driving forces of link formation. Then we describe balance theory to clarify the link sign structure.

Link Formation. The first factor of link formation is homophily [15]. Homophily is characterized 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.

The second factor is preferential attachment [18], which states that nodes with higher connectivity are more likely to form additional links. According to preferential attachment, the link formation likelihood is neither symmetric nor transitive. For example, following celebrity in real world directed networks are not likely to be reciprocated. Instead, likelihood of link formation is asymmetric and determined by distinct connectivity properties of nodes. 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 much 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.

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 $s : E \rightarrow \{-1,+1\}$. We aim to learn embedding function $f : V \rightarrow \mathbb{R}^d$ which depicts link structure including sign and direction. We exploit the random walk based network embedding framework described in Section 2.1; embedding function is derived by maximizing likelihood over node pairs sampled from random walk simulation. 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) **Direction asymmetry.** How can we represent asymmetry in edge direction within symmetric distance model?

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 matches 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 direction asymmetry by distinguishing likelihood formulations of two reciprocal directed edges. Bias terms model not only direction asymmetry but also individual connectivity and affect likelihood as stated in preferential attachment (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, introduce additional optimization techniques, and present social theoretical and 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 target pairs of nodes within a window of size k are selected from the walk sequences. We define sign and direction for each target pair aggregating information along the path.

Sign of the node pair is inferred according to the balance theory as described in Section 2.2: 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 target 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:

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: embedding and context matrix $W, W' \in \mathbb{R}^{|V| \times d}$, 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 and W' 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 ∈ 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 ∈ Walks do
9:      for all (u, v) within distance k in walk do
10:         Update W_u, W'_v, 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' ∈ V
13:             Update W_u and W'_{v'} by taking gradient descent step according to Equation (6)
14:         end for
15:     end for
16: end for
```

$$J = \sum_{(u,v) \in \mathcal{D}} [-\log P(u, v) + \sum_{j=1}^{n} -\log P(u, v_j')] + \frac{\lambda}{2}(\|b^{in,+}\|^2 + \|b^{in,-}\|^2 + \|b^{out,+}\|^2 + \|b^{out,-}\|^2) \quad (3)$$

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:

$$P(u, v) = \begin{cases} \sigma(W_u \cdot W_v' + b_u^{out,+} + b_v^{in,+}) & \textbf{if } sign(u, v) > 0 \\ \sigma(-W_u \cdot W_v' + b_u^{out,-} + b_v^{in,-}) & \textbf{if } sign(u, v) < 0 \\ \sigma(-W_u \cdot W_v') & \textbf{if } v \text{ is a noise} \end{cases} \quad (4)$$

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 and bias terms. In the rest of this section, we elaborate on each component of our likelihood function.

Signed proximity term. The first component of the likelihood function is signed inner product proximity term $\pm W_u \cdot W_v'$. In our model, positively connected nodes are viewed as having proximate embedding vectors; negative and noise pair of nodes are interpreted to be far away from each other in the embedding space. 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.

Bias terms. We employ bias terms $b_u^{out,\pm}$, $b_v^{in,\pm}$ as a second component of the likelihood function in order to model asymmetry. Likelihood function contains two bias terms: out-link bias of a source node and in-link bias of a target node. Source and target are determined by the order 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. 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 $b^{out,-}$.

We train our model using gradient descent optimization. The derivative required to update each parameter one step in gradient descent stage is distinct for target pair and noise pair. For target pair (u, v), two weight vectors W_u, W'_v 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 a target pair (u, v) is as follows:

$$
\begin{aligned}
\frac{\partial J_{(u,v)}}{\partial W_u} &= -sign(u,v)W'_v(1 - P(u,v)) \\
\frac{\partial J_{(u,v)}}{\partial W'_v} &= -sign(u,v)W_u(1 - P(u,v)) \\
\frac{\partial J_{(u,v)}}{\partial b_u^{out,sign(u,v)}} &= -(1 - P(u,v)) + \lambda b_u^{out,sign(u,v)} \\
\frac{\partial J_{(u,v)}}{\partial b_v^{in,sign(u,v)}} &= -(1 - P(u,v)) + \lambda b_v^{in,sign(u,v)}
\end{aligned}
\tag{5}
$$

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 result in diverging magnitude of bias values.

For noise pair (u, v'_j), only two weight vectors W_u, $W'_{v'_j}$ 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:

$$
\begin{aligned}
\frac{\partial J_{(u,v'_j)}}{\partial W_u} &= W'_{v'_j}(1 - P(u, v'_j)) \\
\frac{\partial J_{(u,v'_j)}}{\partial W'_{v'_j}} &= W_u(1 - P(u, v'_j))
\end{aligned}
\tag{6}
$$

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

3.4 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. A node with positive degree one shares all of its neighborhood structure with the only node linked to it. Consequently the learning process performs huge number of redundant calculations for the degree-one node and its only neighbor. 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 [17], 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.

3.5 Social Theoretical Analysis

In this section, we describe social theoretical interpretation of our model. First, 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 2.2: homophily and preferential attachment.

In our signed directed network embedding framework, we utilize signed proximity term so that positively linked nodes are placed closely together and negatively linked nodes are placed far apart. Balance theory is consistent with this distance structure in vector space embedding. 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 cannot be properly placed in the embedding space either. Therefore, our embedding method naturally encourages the embedding vectors to be learned to follow balance theory.

The bias terms model asymmetry in edge direction, 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.

3.6 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 $O(|V|wl)$. □

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 with the following questions.

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

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

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 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) [13]. This model defines two types of hand-engineered features: signed degrees and triad relationship types.
- Node2vec (N2V) [8]. This is a random walk based network embedding method for unsigned network. Embedding is learned on positive subgraph with this method.
- Matrix Factorization (MF) [9]. This method learns low rank structure of signed network by matrix factorization.
- Balanced Normalized Signed Laplacian (BNS) [33]. This method applies a spectral embedding on modified Laplacian matrix which balances positive and negative links.
- Signed Network Embedding (SNE) [32]. This method utilizes log-bilinear model with random walk sampling.
- Signed Network Embedding (SiNE) [27]. 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 [20]: $w = 80$, $l = 40$, 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

Table 4: Performance on prediction of edge sign.

		SIDE (proposed)	FE	N2V	MF	BNS	SNE	SiNE
AUC	Epinions	**0.967**	0.951	0.764	0.920	0.893	0.820	0.860
	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
F1	Epinions	**0.972**	0.960	0.893	0.957	0.948	0.924	0.922
	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

vectors of source and target nodes. We use two methods for combination, element-wise product and concatenation, and report the better result for each dataset. Concatenation always outperforms element-wise product except when predicting Epinions dataset with SIDE. We use 5-fold cross validation. Each training set is used to train both embedding vectors and logistic regression model.

We report *AUC* and $F1-score$ of methods as shown in Table 4. The higher the metrics are, the higher is the predictive accuracy. 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. Note that feature engineering method (FE) is the only comparable competitor and uses features specifically targeting link sign prediction task. N2V, which is learned on positive subgraph, shows the worst accuracy because no sign information is fed in the learning process. Negative edges provide additional information which does not exist in positive subgraph.

4.3 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 [27] because even one epoch takes much larger time than SIDE.

As shown in Figure 2, 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.

4.4 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.

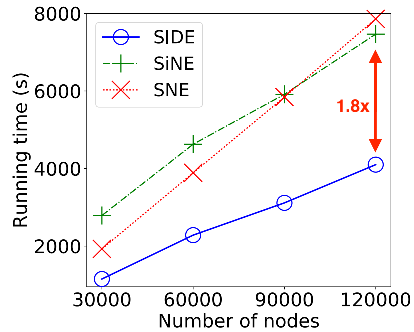

Figure 2: SIDE runs much faster than baseline methods, and shows linear scalability with the smallest slope on the number of nodes.

Table 5: Analysis of Signed proximity terms.

	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

Kolmogorov-Smirnov test is a test on the equality of two probability distribution. Smaller magnitude of p-values for both tests 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 standard deviations.

We verify the preferential attachment interpretation of the learning process of our bias terms. Bias terms might not be directly related to corresponding degree. For example, high positive degree might induce many negative indirect connections. On the other hand, direction is not inverted in multi-step connections. Therefore, we analyze unsigned, directed degree and bias terms. We demonstrate that the bias values are proportional to the degree of corresponding nodes. We divide nodes into 10 buckets according to degree deciles. Figure 3 shows how average bias values are different for each bucket. As node degree increases, the expected value of bias term increases.

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

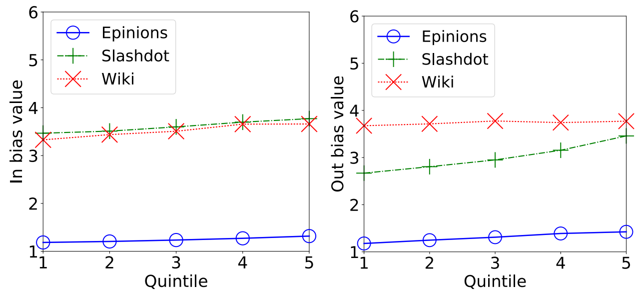

Figure 3: Bias value increases as the degree quintile increses. This is consistent with the preferential attachment interpretation of bias terms.

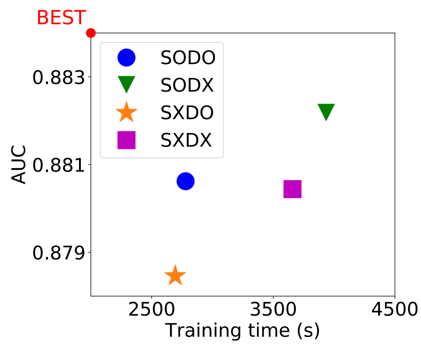

Figure 4: The tradeoff between training time and performance of SIDE as optimization techniques are activated.

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, 8, 10, 19, 20, 24–26, 30, 31]. 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. [19] 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. [25] leveraged stacked autoencoder to reconstruct normalized similarity matrix. Wang et al. [26] 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. [20] first suggested the framework of random walk based network embedding method. Grover and Leskovec [8] 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 [12, 33]. These methods, however, are applicable only to undirected networks and are not scalable because of the complexity of eigendecomposition. Wang et al. [27] 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. [32] 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 asymmetry within symmetric metric space, 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. We also propose techniques to optimize truncated random walk based network embedding frameworks. Social theoretical analysis shows the explainability of our model components. We prove that the complexity of our method is linear in the number of nodes. Experimental results show that SIDE well represents sign and direction into embedding space, outperforms competitors in link sign prediction task, and is learned efficiently.

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] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.
[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] Zhaoming Wu, Charu C Aggarwal, and Jimeng Sun. 2016. The troll-trust model for ranking in signed networks. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, 447–456.

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

[31] 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.

[32] 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.

[33] 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.