SIGNet: Scalable Embeddings for Signed Networks

Mohammad Raihanul Islam *1,2, B. Aditya Prakash†1,2 and Naren Ramakrishnan‡1,2

1Department of Computer Science, Virginia Tech, VA, USA
2Virginia Tech, Discovery Analytics Center, Arlington, VA 22203

Abstract

Recent successes in word embedding and document embedding have motivated researchers to explore similar representations for networks and to use such representations for tasks such as edge prediction, node label prediction, and community detection. Existing methods are largely focused on finding distributed representations for unsigned networks and are unable to discover embeddings that respect polarities inherent in edges. We propose SIGNet, a fast scalable embedding method suitable for signed networks. Our proposed objective function aims to carefully model the social structure implicit in signed networks by reinforcing the principles of social balance theory. Our method builds upon the traditional word2vec family of embedding approaches but we propose a new targeted node sampling strategy to maintain structural balance in higher-order neighborhoods. We demonstrate the superiority of SIGNet over state-of-the-art methods proposed for both signed and unsigned networks on several real world datasets from different domains. In particular, SIGNet offers an approach to generate a richer vocabulary of features of signed networks to support representation and reasoning.

1 Introduction

Social and information networks are ubiquitous today across a variety of domains; as a result, a large body of research has developed to help construct discriminative and informative features for network analysis tasks such as classification [3], prediction [21], visualization [30], and entity recommendation [10]. Classical approaches to find features and embeddings are motivated by dimensionality reduction research and extensions, e.g., approaches such as Laplacian eigenmaps [2], non-linear dimension reduction [28, 27], graph embedding strategies [8], kernel methods [31], and spectral embedding [17, 33]. More recent research has focused on developing network analogies to distributed vector representations such as word2vec [22, 23]. In particular, by viewing sequences

*raihan8@cs.vt.edu
†badityap@cs.vt.edu
‡naren@cs.vt.edu
of nodes encountered on random walks as documents, methods such as DeepWalk [24], node2vec [13], and LINE [6] learn similar representations for nodes (viewing them as words).

Although these approaches are scalable to large networks, they are primarily applicable to only unsigned networks. Signed networks are becoming increasingly important in online media, trust management, and in law/criminal applications. As we will show, applying the above methods to signed networks results in key information loss in the resulting embedding. For instance, if the sign between two nodes is negative, the resulting embeddings could place the nodes in close proximity, which is undesirable.

A recent attempt to fill this gap is the work of Wang et al. [32] wherein the authors learn node representations by optimizing an objective function through a multi-layer neural network based on structural balance theory. This work, however, models only local connectivity information through 2-hop paths and fails to capture global balance structures prevalent in a network.

Our contributions are:

1. We propose SIGNet, a scalable node embedding method for feature learning in signed networks that maintains structural balance in higher order neighborhoods. SIGNet is very generic by design, and can handle both directed and undirected networks, including weighted or unweighted (binary) edges.

2. We propose a novel node sampling method as an improvement over conventional negative sampling. The basic idea is to maintain a cache of nodes during optimization integral for maintaining the principles of structural balance in the network. This targeted node sampling can be treated as an extension of the negative sampling used in word2vec models.

3. Through extensive experimentation, we demonstrate that SIGNet generates better features suitable for a range of prediction tasks such as edge prediction and node label prediction. SIGNet is able to scalably generate embeddings for networks with millions of nodes.

2 Problem Formulation

Definition 1. Signed Network: A signed network can be defined as \( G = (V, E) \), where \( V \) is the set of vertices and \( E \) is the set of edges between the vertices. Each element \( v_i \) of \( V \) represents an entity in the network and each edge \( e_{ij} \in E \) is a tuple \((v_i, v_j)\) associated with a weight \( w_{ij} \in \mathbb{Z} \). The absolute value of \( w_{ij} \) represents the strength of the relationship between \( v_i \) and \( v_j \), whereas the sign represents the nature of relationship (e.g., friendship or antagonism). A signed network can be either directed or undirected. If \( G \) is undirected then the order of vertices is not relevant (i.e. \((u, v) \equiv (v, u)\)). On the other hand if \( G \) is directed then order becomes relevant (i.e. \((u, v) \neq (v, u)\) and \( w_{ij} \neq w_{ji} \)).

Because the weights in a signed network carry a combined interpretation (sign denotes polarity and magnitude denotes strength), conventional proximity assumptions used in unsigned network representations (e.g., in [6] [13]) cannot be applied for signed networks. Consider a network wherein the nodes \( i \) and \( j \) are positively connected and the nodes \( k \) and \( i \) are negatively connected (see Fig. 1)
Figure 1: Given a signed network (a), a conventional network embedding (b) will not take signs into account and can result in faulty representations. (c) SIGNet learns embeddings respecting sign information between edges.

(a)). Suppose the weights of the edges $e_{ij}$ and $e_{ik}$ are $+w_{ij}$ and $-w_{ik}$ respectively. Now if $|+w_{ij}| < |-w_{ik}|$, conventional embedding methods will place $i$ and $k$ closer than $i$ and $j$ owing to the stronger influence of the weight (Fig. 1(b)). Ideally, we would like a representation wherein nodes $i$ and $j$ are closer than nodes $i$ and $k$, as shown in Fig. 1(c). This example shows that modeling the polarity of the relationship is as important as modeling the strength of the relationship.

Figure 2: Signed triangles in an undirected graph. (a) and (b) are balanced but (c) and (d) are not.

To accurately model the interplay between the vertices in signed networks we use the theory of *structural balance* proposed by Heider [10]. Structural balance theory posits that triangles with an odd number of positive edges are more plausible than an even positive edges (see Fig. 2). Cartwright and Harary [4] proposed a theorem about global network structure interpreted via balance theory. This theorem asserts that in an undirected network where each triangle is connected and obeys the structural balance theory, the vertices can be partitioned into two disjoint subsets where intra-subset edges are positive and inter-subset edges are negative:
Theorem 1. (Cartwright and Harary, 1956): A signed network is balanced if and only if the vertices can be divided into two mutually exclusive subsets so that each positive edge connects nodes within the subset and each negative edge connects nodes between the subsets.

Although structural balance theory has been adapted and refined significantly (e.g., by Davis [5]), here we primarily focus on the original notion of structural balance. Theorem 1 suggests that in a perfectly balanced network we can observe two groups, where members within groups are friendly to each other, and members across the groups are not. Inspired by this observation, we aim to obtain embeddings for signed networks that reflect such a dichotomy.

Problem Statement: Scalable Embedding of Signed Networks (SIGNet):
Given a signed network \( G \), compute a low-dimensional vector \( d_i \in \mathbb{R}^K \), \( \forall i \in V \), where positively related vertices reside in close proximity and negatively related vertices are distant.

3 Scalable Embedding of Signed Networks (SIGNet)

3.1 SIGNet for undirected networks

Consider a weighted signed network defined as in Section 2. Now suppose each \( v_i \) is represented by a vector \( x_i \in \mathbb{R}^K \). Then a natural way to compute the proximity between \( v_i \) and \( v_j \) is by the following function (ignoring the sign for now):

\[
p_u(v_i, v_j) = \sigma(x_j^T \cdot x_i) = \frac{1}{1 + \exp(-x_j^T \cdot x_i)}
\]

where \( \sigma(a) = \frac{1}{1 + \exp(-a)} \). Now let us breakdown the weight of edge \( w_{ij} \) into two components: \( r_{ij} \) and \( s_{ij} \). \( r_{ij} \in \mathbb{R}^+ \) represents the absolute value of \( w_{ij} \) (i.e., \( r_{ij} = |w_{ij}| \)) and \( s_{ij} \in \{-1, 1\} \) represents the sign of \( w_{ij} \). Given this breakdown of \( w_{ij} \) and incorporating the weight information, the objective function for undirected signed network can be written as:

\[
O_{un} = \sum_{e_{ij} \in E} r_{ij} \times \sigma(s_{ij}(x_j^T \cdot x_i))
\]

By maximizing Eqn. (2) we obtain a vector \( x_i \) of dimension \( K \) for each node \( v_i \in V \) (we will also use \( d_i \) to denote this embedding, for reasons that will become clear in the next section).

3.2 SIGNet for directed networks

Computing embeddings for directed networks is trickier due to the asymmetric nature of neighborhoods (and thus, contexts). For instance, if the edge \( e_{ij} \) is positive, but \( e_{ji} \) is negative, it is not clear if the respective representations for nodes \( i \) and \( j \) should be proximal or not. We solve this problem by treating each vertex as each vertex as itself plus a specific context; for instance, a positive edge \( e_{ij} \) is interpreted to mean that given the context of node \( j \), node \( i \) wants to be closer. This enables us to treat all nodes consistently without worrying about
reciprocity relationships. To this end, we introduce another vector \( y_i \), \( \forall i \in V \) in addition to \( x_i \). For a directed edge \( e_{ij} \) the probability of context \( v_j \) given \( v_i \) is:

\[
p_d(v_j|v_i) = \frac{\exp(s_{ij}(y_j^T \cdot x_i))}{\sum_{k=1}^{|V|} \exp(s_{ik}(y_k^T \cdot x_i))}
\]

Treating the same entity as itself and as a specific context is very popular in the text representation literature \[22, 23\]. The above equation defines a probability distribution over all context space w.r.t. node \( v_i \). Now our goal is to optimize the above objective function for all the edges in the network. However we also need to consider the weight of each edge in the optimization. Incorporating the absolute weight of each edge we obtain the objective function for a directed network as:

\[
O_{\text{dir}} = \sum_{e_{ij} \in E} r_{ij} p_d(v_j|v_i)
\]

By maximizing the above function we will obtain two vectors \( x_i \) and \( y_i \) for each \( i \in V \). The vector \( x_i \) models the outward connection of a node whereas \( y_i \) models the inward connection of the node. Therefore the concatenation of \( x_i \) and \( y_i \) represents the final embedding for each node. We denote the final embedding of node \( i \) as \( d_i \). It should be noted that for undirected network \( d_i = x_i \) whereas for a directed network \( d_i \) is the concatenation of \( x_i \) and \( y_i \). This means \( |x_i| = |y_i| = \frac{K}{2} \) in the case of directed graph (for the same representational length).

### 3.3 Efficient Optimization through Targeted Node Sampling

The denominator of Eqn. 3 is very hard to compute as we have to marginalize the conditional probability over the entire vertex set \( V \). We adopt the classical negative sampling approach \[24\] wherein negative examples are selected randomly from some distribution for each edge \( e_{ij} \). Incorporating the negative sampling we obtain the following objective function for each edge \( e_{ij} \):

\[
\log[\sigma(s_{ij}(y_i^T \cdot x_i))] + \sum_{c=1}^{N} E_{v_n \sim P_n(v)} \log[\sigma(-(y_n^T \cdot x_i))] \]

Here \( N \) is the number of negative examples per edge. However, for signed network conventional negative sampling does not work. For example consider the network from Fig. 3(a). Viewing this example as an unsigned network, while optimizing for edge \( e_{uv} \) we will consider \( u \) and \( y \) as negative examples and thus they will be placed distantly from each other. However, in a signed network context, \( u \) and \( y \) have a friendlier relationship (than with, say, \( x \)) and thus should be placed closer to each other. We propose a new sampling approach, referred to as simply targeted node sampling wherein we first create a cache of nodes for each node with their estimated relationship according to structural balance theory and then sample nodes accordingly.

#### 3.3.1 Constructing the cache for each node

We aim to construct a cache of positive and negative examples for each node \( i \) where the positive example cache \( \eta^+_i \) contains nodes which should have a
Figure 3: (a) depicts a small network to illustrate why conventional negative sampling does not work. \( u \) and \( y \) might be considered too distant for their representations to be placed close to each other. Targeted node sampling solves this problem by constructing a cache of nodes which can be used as sampling. (b) shows how we resolve conflicts. Although there are two ways to proceed from node \( a \) to \( d \) the shortest path is \( a, b, c, d \), which estimates a net positive relation between \( a \) and \( d \). As a result \( d \) will be added to \( \eta^+_{a} \). However for node \( e \) there are two shortest paths from \( a \), with the path \( a, h, g, f, e \) having more positive edges but with a net negative relation, so \( e \) will be added to \( \eta^-_{a} \).

positive relationship with \( i \) and the negative example cache \( \eta^-_{i} \) contains nodes which should have negative relationship with \( i \), according to structural balance theory. To construct these caches for each node \( i \), we apply random walks of length \( l \) starting with \( i \) to obtain a sequence of nodes. Suppose the sequence is \( \Omega = \langle i, v_{n_0}, \ldots, v_{n_{l-1}} \rangle \). Now we add each node \( v_{n_k} \) to either \( \eta^+_{i} \) or \( \eta^-_{i} \) by observing the estimated sign between \( i \) and \( v_{n_k} \). The estimated sign is computed using the following formula:

\[
\tilde{s}_{iv_{n_k}} = \tilde{s}_{i,v_{n_k-1}} \times s_{v_{n_k-1},v_{n_k}}
\]

Here \( \tilde{s}_{i,v_{n_k-1}} \) is the estimated sign between node \( i \) and node \( v_{n_k-1} \). (In the rare instances where the sign (i.e. weight) between two nodes is zero, we consider them as negative). If node \( v_k \) is not a neighbor of node \( i \) and \( \tilde{s}_{iv_{n_k}} \) is positive then we add \( v_k \) to \( \eta^+_{i} \). On the other hand if \( \tilde{s}_{iv_{n_k}} \) is negative and \( v_k \) is not a neighbor of \( i \) then we add it to \( \eta^-_{i} \). For example for the graph shown in Fig. 3 (a), suppose a random walk starting with node \( u \) is \( \langle u, v, w, z \rangle \). Here node \( w \) will be added to \( \eta^+_{u} \) because \( \tilde{s}_{uw} = s_{uv} \times s_{vw} > 0 \) and \( w \) is not a neighbor of \( u \). On the other hand, \( z \) will be added to node \( \eta^-_{u} \) since \( \tilde{s}_{uz} = s_{uw} \times s_{uw} <= 0 \) and \( z \) is not a neighbor of \( u \).

The one problem with this approach is that a node \( b \) may be added to both \( \eta^+_{a} \) and \( \eta^-_{a} \). We define the reason for this conflict in Theorem 2. We resolve this situation by computing the shortest path between \( a \) and \( b \) and compute \( \tilde{s}_{ab} \) between them using the shortest path, then add to either \( \eta^+_{a} \) or \( \eta^-_{b} \) based on \( \tilde{s}_{ab} \). To compute the shortest path we have to consider the network as unsigned since negative weight has a different interpretation for shortest path algorithms. If there are multiple shortest paths with equal length then at most one will have the highest number of positive edges (Theorem 3). We pick this path to compute \( \tilde{s}_{ab} \). A scenario is shown in Fig. 3 (b).
Theorem 2. Node $v$ will be added to both $\eta_u^+$ and $\eta_u^-$ if the path from node $u$ to $v$ has at least one unbalanced cycle.

Proof. (By contradiction.) Suppose there is a conflict for node $u$ where $\eta_u^+$ and $\eta_u^-$ both contain node $v$. Since there are two distinct $u$-$v$ paths the network will contain a cycle $c$. Now it is evident that the common edges of both paths are not responsible for the conflict since they occur in both paths. Now if cycle $c$ is balanced there will be an even number of negative edges which will be distributed between the distinct $x$-$y$ paths in $c$. The distribution can occur in two ways: either both paths will have an odd number of negative edges or an even number of negative edges. In both cases the estimated sign between the $x$-$y$ paths will be the same. However, this is a contradiction because the final estimated sign of two $u$-$v$ paths are different and the signs between the common path are same, so the signs between the $x$-$y$ paths must be different. Therefore, cycle $c$ cannot be balanced and hence contains an odd number of negative edges. Thus we have identified at least one unbalanced cycle.

Theorem 3. If there are multiple equal length shortest paths from $u$ to $v$ then at most one path will have the largest number of positive edges.

Proof. From Theorem 2 we know that the $u$-$v$ paths will contain at least one unbalanced cycle $c$. Now paths are equal in length so the paths on the cycle will also be equal in length. Suppose paths on the cycle $c$ are $x$-$y$. Now since $c$ is unbalanced it will have an odd number of edges. Since the number of edges are odd, at least one path will have one less negative edge than the others. Therefore one path will have more positive edges than all the other paths.

3.3.2 Targeted edge sampling during optimization

Now after constructing the cache $\eta_i = \eta_i^+ \cup \eta_i^-$ for each node $i$, we can apply the targeted sampling approach for each node. Here our goal is to extend the objective of negative sampling from classical word2vec approaches [23]. In traditional negative sampling, a random word-context pair is negatively sampled for each observed word-context pair. In a signed network both positive and negative edges are present, and thus we aim to conduct both types of sampling while sampling an edge observing its sign. Therefore when sampling a positive edge $e_{ij}$, we aim to sample multiple negative nodes from $\eta_i^-$ and while sampling a negative edge $e_{ik}$ we aim to sample multiple positive nodes from $\eta_i^+$. Therefore the objective function for each edge becomes:

$$C_{ij} = \log[\sigma(s_{ij}(y_j^T \cdot x_i))] + \sum_{c=1}^{N} v_n \sim \tau(s_{ij}) \log[\sigma(\tilde{s}_{in}(y_n^T \cdot x_i))]$$

(7)

Here $\tau$ is a function which selects from $\eta_i^+$ or $\eta_i^-$ based on the sign $\tilde{s}_{in}$. $\tau$ selects from $\eta_i^+$ if $\tilde{s}_{in} \leq 0$ or returns $\eta_i^-$ if $\tilde{s}_{in} > 0$.

The benefit of targeted node sampling in terms of global balance considerations across the entire network is shown in Fig. 4. Here we compare how our proposed approach SIGNet and SiNE [32] maintain structural balance. For simplicity suppose only edge $e_{ij}$ has a negative sign. Now SiNE only optimizes w.r.t. pairs of edges in 2-hop paths each having different signs. Therefore optimizing the edge $e_{ij}$ involves only the immediate neighbors of node $i$ and $j$, i.e. $l, m, n, o$. 


Figure 4: A comparative scenario depicting the optimization process inherent in both SiNE (a) and SIGNet (b). The shaded vertices represent the nodes both methods will consider while optimizing the edge \( e_{ij} \). We can see the SiNE only considers the immediate neighbors because it optimizes edges in 2-hop paths having opposite signs. On the other hand, SIGNet considers higher order neighbors \((a, b, c, x, y, z)\) for targeted node sampling. However, SIGNet skips the immediate neighbors while it uses higher order neighbors \((i, e, l, m, n)\). Note that SIGNet actually uses immediate neighbors as separate examples \((i.e., e_{il}, e_{in}, etc.)\). In this manner, SIGNet covers more nodes to optimize the embedding space than SiNE.

3.4 Discussion

We now discuss several computational aspects of the SIGNet model. **Optimization:** We adopt the asynchronous stochastic gradient method (ASGD) \([26]\) to optimize the objective function \( \mathcal{O}_{ij} \) for each edge \( e_{ij} \). The ASGD method randomly selects a mini batch of randomly selected edges and update embeddings at each step. Now for each edge \( e_{ij} \) the gradient of the objective function will have a constant coefficient \( r_{ij} \) \((i.e., |w_{ij}|)\). Now if the absolute weights of the edges have a high variance, it is hard to find a good learning rate. For example, if we set the learning rate very small it would work well for large weighted edge but for small weighted edges the overall learning will be very inadequate resulting in poor performance. On the other hand, a large learning rate will work well for edges with smaller weights but for edges with large weight the gradient will be out of limits. To remedy this we adopt the *edge sampling* used in \([6]\). In edge sampling all the weighted edges treated as binary edges with non-negative weights \((i.e., \text{absolute value of edges } r_{ij})\). Now the edges are sampled during optimization according to the multinomial distribution constructed from the absolute value of the edge weights. For example, suppose all the absolute values of the edges are stored in the set \( R = \{r_1, r_2, \ldots, r_{|E|}\} \). Now during the optimization each edge is sampled according to the multinomial distribution constructed from \( R \). However, each sampling from \( R \) would take \( O(E) \) time, which is computationally expensive for large network. To remedy this we use...
Algorithm 1 The SIGNet algorithm

Input: (Graph $G = (V, E)$, embedding size $K$, walks per node $r$, walk length $l$, total number of samples $s$, initial learning rate $\gamma$)

Output: $d_k \in \mathbb{R}^K, \forall k \in V$

1. for all $n \in V$ do
2. \hspace{1em} for $i = 1$ to $r$ do
3. \hspace{2em} $\omega_{ni} = \text{RandomWalk}(G, n, l)$
4. for all $n \in V$ do
5. \hspace{1em} for $i = 1$ to $r$ do
6. \hspace{2em} for each $v \in \omega_{ni}$ do
7. \hspace{3em} Estimate relation between $v$ and $n$ using Eqn. 6
8. \hspace{3em} Add $v$ to either $\eta^+_n$ or $\eta^-_n$ based on the relation
9. repeat
10. \hspace{1em} for each mini-batch of edges do
11. \hspace{2em} Sample an edge $e_{ij}$ using edge sampling method
12. \hspace{2em} Optimize the objective function in Eqn. 7
13. \hspace{2em} Update learning rate $\gamma$
14. until in total $s$ samples are processed

the alias table approach proposed in [19]. An alias table takes $O(1)$ time while continuously drawing samples from a constant discrete multinomial distribution.

**Threshold value for $\eta_i$:** Theoretically there should not be any bound on the size of $\eta^+_i$ and $\eta^-_i$. However empirical analysis shows limiting the size of $\eta^+_i$ to very small values (i.e. 5 – 7) actually gives better results.

**$\eta_i$ for low degree nodes:** Nodes with a low degree may not have an adequate number of samples for $\eta^+_i$ and $\eta^-_i$ from the random walks. This is why it is possible to exchange the nodes within $\eta^+_i$ and $\eta^-_i$. For example if node $x \in \eta^+_i$, one can add node $i$ to $\eta^+_x$. The same approach can be explored for $\eta^-_i$ in case there is an inadequate number of samples. However, we advise caution for taking this approach since it may violate structural balance for directed network.

**Embedding for new vertices:** SIGNet can learn embedding for newly arriving vertices. Since this is a network model, we can assume that advent of new vertices means we know its connection with existing nodes (i.e., neighbors). Suppose the new vertex is $n$ and its set of neighbors is $N_n$. We just have to construct $\eta_n$ and optimize the newly formed edges using the same optimization function stated in Eqn. 7 to obtain the embedding of node $n$.

**Complexity:** Constructing $\eta_i$ for node $i$ takes $O(rl)$ time where $l$ is the length of random walk and $r$ is the number of walk for each node. Since $rl \ll |V|$, the total cache construction actually takes very little time w.r.t. vertex size. Moreover conflict resolution only takes place for very rare instances where the length of the shortest path is at most $l$. This cost is thus negligible compared to random walk and cache construction time. Now, for optimizing each edge along with the node sampling take $O(K(N + 1))$, where $K$ is the size of embedding space and $N$ is the size of node sampling. The total complexity of optimization then become $O(K(N + 1)|E|)$, where $E$ is the set of edges. Therefore the overall complexity becomes $O(rl|V| + K(N + 1)|E|)$. A pseudocode of SIGNet is shown in Algorithm 1.
4 Experiments

In this section we present our empirical evaluation of SIGNet compared to other state-of-the-art methods, with a view toward answering the following questions:

1. Are the node embeddings learned by SIGNet interpretable? (Section 4.2)
2. Does the embedding space learned by SIGNet support structural balance theory? (Section 4.3)
3. Are representations learned by SIGNet effective at edge label prediction? (Section 4.4)
4. Are representations learned by SIGNet effective at node label prediction? (Section 4.5)
5. How much more effective is our sampling strategy in the presence of partial information? (Section 4.6)
6. How scalable is SIGNet for large networks? (Section 4.7)
7. Do parameter variations in SIGNet lead to overfitting? (Section 4.8)

4.1 Experimental Setup

We compare our algorithm against both the state-of-the-art method proposed for signed and unsigned network embedding. The description of the methods are below:

- **node2vec [13]**: This method, not specific to signed networks, computes embeddings by optimizing the neighborhood structure using informed random walks.

- **SiNE [32]**: This method uses a multi-layer neural network to learn the embedding by optimizing an objective function satisfying structural balance theory. SiNE only concentrates on the immediate neighborhood of vertices rather than on the global balance structure.

- **SIGNet-NS**: This method is similar to our proposed method SIGNet except it uses conventional negative sampling instead of our proposed targeted node sampling.

- **SIGNet**: This is our proposed SIGNet method which uses random walks to construct a cache of positive and negative examples for targeted node sampling.

In the discussion below, we focus on four real world signed network datasets (see Table 1). Out of these four, two datasets are from social network platforms—Epinions and Slashdot—courtesy the Stanford Network Analysis Project (SNAP). The details on how the signed edges are defined are available at the project website[^1]. The third dataset we study is a citation network we constructed from written case opinions of the Supreme Court of the United States (SCOTUS). We expand the notion of SCOTUS citation network [12] into a signed network.

[^1]: http://snap.stanford.edu/
Table 1: Statistics of the datasets used for performance evaluation. In social network datasets negative edges are underrepresented, however in ADJNet and SCOTUS they are well represented. ADJNet and SCOTUS also contain binary labels.

| Statistics       | Epinions | Slashdot | ADJNet | SCOTUS |
|------------------|----------|----------|--------|--------|
| total nodes      | 131828   | 82144    | 4579   | 28305  |
| positive edges   | 717667   | 425072   | 10708  | 43781  |
| negative edges   | 123705   | 124130   | 7044   | 42102  |
| total edges      | 841372   | 549202   | 17752  | 85883  |
| % negative edges | 14.703   | 22.602   | 39.680 | 49.023 |

To understand this network, it is important to note that there are typically two main parts to a SCOTUS case opinion. The first part contains the majority and any optional concurring opinions where justices cite previously argued cases to defend their position. The second part (optional, does not exist in a unanimous decision) consists of dissenting opinions containing arguments opposing the decision of the majority opinion. In our modeling, nodes denote cases (not opinions). The citation of one case’s majority opinion to another case will form a positive relationship, and citations from dissenting opinions will form a negative relationship. We collected all written options from the inception of SCOTUS to construct the citation network. Moreover, we also collected the decision direction of supreme court cases from The Supreme Court Database. This decision direction denotes whether the decision is conservative or liberal, information that we will use for validation. The last dataset (ADJNet) is an adjective network constructed from the synonyms and antonyms collected from Wordnet database. Label information about whether the adjective is positive or negative comes from SentiWordNet.

Unless otherwise stated, we set the dimension of $x_i$ and $y_i$ to 20 for both SIGNet-NS and SIGNet. Therefore the final embedding for each node becomes 40. For a fair comparison, the embedding dimension for node2vec and SiNE is set to 40. We all set the total number of samples (examples) to 100 million and $N = 5$ for SIGNet-NS and SIGNet. Moreover, we set $l = 50$ and $r = 1$ for SIGNet. For all the other parameters for node2vec and SiNE we use the settings recommended in their respective papers.

4.2 Are Embeddings Interpretable?

For visual depiction of embeddings, we first utilize a small dataset denoting relations between sixteen tribes in Central Highlands of New Guinea. This is a signed network showing the alliance and hostility between the tribes. We learned the embeddings in two dimensional space as an undirected network as shown in Fig. 5. We can see that in general solid blue edges (alliance) are shorter than the dashed red edges (hostility) confirming that allied tribes are closer than the hostile tribes. One notable point is tribe MASIL has no enemies and often works as a peace negotiator between the tribes. We can see that

2http://scdb.wustl.edu/
3http://sentiwordnet.isti.cnr.it/
MASIL positions nicely between two groups of tribes \{OVE, GAHUK, ASARO, UKUDŽ, ALIKA, GEHAM\} and \{UHETO, SEUVE, NAGAM, KOHIK, NOTOH\}. The tribes within these two groups are only allied to each other and MASIL but they are hostile to other tribes belonging to different groups. This actually justifies the position of MASIL. As reported in [14] there is another such group which consists of the tribes \{NAGAD, KOTUN, GAMA, GAVEV\}; notice that they position themselves in the lower left corner far away from other two groups. Therefore the embedding space learned by SIGNet clearly depicts alliances and relationships among the tribes.

![Figure 5: Two dimensional embedding of alliances among sixteen tribes of New Guinea. Alliance between the tribes is shown in solid blue edges and a hostile relation is shown in dashed red edges. We can see that edges representing alliance are comparatively shorter than the edges represents hostility.](image)

4.3 Analyzing the Embedding Space

Here we present our analysis on whether the embedding space learned by SIGNet follows the principles of structural balance theory. We calculate the mean Euclidean distance between representations of nodes connected by positive versus negative edges, as well as their standard deviations (see Table 2). The lower value of positive edges suggests positively connected nodes stay closer together than the negatively connected nodes indicating that SIGNet has successfully learned the embedding using the principles of structural balance theory. Moreover, the ratio of average distance between the positive and negative edges is at most 61% over all the datasets suggesting that SIGNet grasps the principles very effectively.
Table 2: Average Euclidean distance between node representations connected by positive edges versus negative edges with std. deviation. We can see that the avg. distance between positive edge is significantly lower than negative edges indicating that SIGNet preserves the conditions of structural balance theory.

| Type of edges | Epinions | Slashdot | SCOTUS | ADJNet |
|---------------|----------|----------|--------|--------|
| positive      | 0.856 ± 0.370 | 0.977 ± 0.306 | 0.835 ± 0.247 | 0.709 ± 0.161 |
| negative      | 1.636 ± 0.228 | 1.600 ± 0.188 | 1.655 ± 0.209 | 1.765 ± 0.075 |
| ratio         | 0.5232 | 0.6106 | 0.5045 | 0.4017 |

4.4 Edge Label Prediction

We now explore the utility of SIGNet for edge label prediction. For all the datasets we sample 80% of the edges as a training set to learn the node embedding. Then we train a logistic regression classifier using the embedding as features and the sign of the edges as label. This classifier is used to predict the sign of the remaining 20% of the edges. Since edges involve two nodes we explore several scores to compute the features for edges from the node embedding. They are described below:

1. **Concatenation**:  \( f_{ij} = d_i \oplus d_j \)
2. **Average**:  \( f_{ij} = \frac{d_i + d_j}{2} \)
3. **Hadamard**  \( f_{ij} = d_i \ast d_j \)
4. \( L_1: f_{ij} = |d_i - d_j| \)
5. \( L_2: f_{ij} = |d_i - d_j|^2 \)

Here \( f_{ij} \) is the feature vector of edge \( e_{ij} \) and \( d_i \) is the embedding of node \( i \). Except for the method of concatenation (which has a feature vector dimension of 80) other methods use 40-dimensional vectors. Since the datasets are typically imbalanced we use the micro-F1 and AUC scores to evaluate our method. (see Table 3). Some key observations from this table are as follows:

1. SIGNet, not surprisingly, outperforms node2vec across all datasets. For datasets that contain relatively fewer negative edges (e.g., 14% for Epinions and 22% for Slashdot), the improvements are modest (around 7%). For ADJNet and SCOTUS where the sign distribution is less skewed, SIGNet outperforms node2vec by a huge margin (19% for ADJNet and 53% for SCOTUS).

2. SIGNet demonstrates a consistent advantage over SiNE, with improvements ranging from 7% (for the social network datasets) to 20 – 30% (for ADJNet and SCOTUS).

3. SIGNet also outperforms SIGNet-NS in almost all scenarios demonstrating the effectiveness of targeted node sampling over negative sampling.
Table 3: Comparison of edge label prediction in all four datasets. We show the micro F1 and AUC score for each feature scoring method. The best F1 score and AUC score across all the scoring method is shown in larger boldface. SIGNet outperforms node2vec and SiNE in every case.

| Eval. | Score | Algorithms | Epinions | Slashdot | ADJNet | SCOTUS |
|-------|-------|------------|----------|----------|---------|--------|
|       | concat | node2vec   | 0.8413   | 0.7765   | 0.5938  | 0.5128 |
|       |       | SINE       | 0.8528   | 0.7737   | 0.5977  | 0.6071 |
|       |       | SIGNet-NS  | 0.9110   | 0.7926   | 0.5993  | 0.5602 |
|       |       | SIGNet     | **0.9197**| **0.8323**| 0.5730  | 0.5570 |
| F1    | avg   | node2vec   | 0.8531   | 0.7746   | 0.6028  | 0.5162 |
|       |       | SINE       | 0.8528   | 0.7735   | 0.5990  | 0.6072 |
|       |       | SIGNet-NS  | 0.8370   | 0.7744   | 0.6199  | 0.5991 |
|       |       | SIGNet     | 0.8791   | 0.8085   | 0.5744  | 0.5116 |
|       | had   | node2vec   | 0.8529   | 0.7732   | 0.6004  | 0.5116 |
|       |       | SINE       | 0.8522   | 0.7736   | 0.5982  | 0.6070 |
|       |       | SIGNet-NS  | 0.8456   | 0.7568   | 0.7054  | **0.7927**|
|       |       | SIGNet     | 0.8831   | 0.7815   | **0.7215**| 0.7917 |
|       | l1    | node2vec   | 0.8523   | 0.7740   | 0.6002  | 0.5086 |
|       |       | SINE       | 0.8521   | 0.7734   | 0.5976  | 0.6071 |
|       |       | SIGNet-NS  | 0.8514   | 0.7636   | 0.6387  | 0.7226 |
|       |       | SIGNet     | 0.9009   | 0.7870   | 0.7028  | 0.7232 |
|       | l2    | node2vec   | 0.8516   | 0.7739   | 0.6007  | 0.5091 |
|       |       | SINE       | 0.7874   | 0.7732   | 0.5978  | 0.6069 |
|       |       | SIGNet-NS  | 0.8481   | 0.7631   | 0.6586  | 0.7418 |
|       |       | SIGNet     | 0.9034   | 0.8093   | 0.7158  | 0.7447 |
|       | gain over node2vec | 7.8068 | 7.1861 | 19.6914 | 53.3708 |
|       | gain over SiNE   | 7.8447 | 7.5740 | 20.4508 | 30.3854 |
|       | gain over SIGNet-NS | 0.9550 | 5.0088 | 2.2824 | -0.1262 |
|       | concat | node2vec   | 0.7734   | 0.7193   | 0.5889  | 0.5160 |
|       |       | SINE       | 0.5600   | 0.5474   | 0.5128  | 0.5196 |
|       |       | SIGNet-NS  | 0.8508   | 0.7805   | 0.6062  | 0.5555 |
|       |       | SIGNet     | **0.9180**| **0.8028**| 0.5614  | 0.5545 |
| AUC   | avg   | node2vec   | 0.7447   | 0.7146   | 0.6070  | 0.5174 |
|       |       | SINE       | 0.5467   | 0.5391   | 0.5277  | 0.5166 |
|       |       | SIGNet-NS  | 0.7328   | 0.7425   | 0.6747  | 0.5368 |
|       |       | SIGNet     | 0.8764   | 0.8230   | 0.5710  | 0.5955 |
|       | had   | node2vec   | 0.7131   | 0.6691   | 0.5386  | 0.5135 |
|       |       | SINE       | 0.5396   | 0.5364   | 0.5262  | 0.5154 |
|       |       | SIGNet-NS  | 0.8193   | 0.6913   | 0.7775  | 0.8798 |
|       |       | SIGNet     | 0.9014   | 0.7737   | **0.7995**| **0.8809**|
|       | l1    | node2vec   | 0.6743   | 0.5631   | 0.5252  | 0.5094 |
|       |       | SINE       | 0.5917   | 0.5156   | 0.5249  | 0.5074 |
|       |       | SIGNet-NS  | 0.7881   | 0.6210   | 0.6789  | 0.7799 |
|       |       | SIGNet     | 0.8781   | 0.7695   | 0.7736  | 0.7814 |
|       | l2    | node2vec   | 0.6717   | 0.5618   | 0.5413  | 0.5075 |
|       |       | SINE       | 0.5601   | 0.5192   | 0.5259  | 0.5089 |
|       |       | SIGNet-NS  | 0.7906   | 0.6256   | 0.6936  | 0.7970 |
|       |       | SIGNet     | 0.9039   | 0.7803   | 0.7882  | 0.8002 |
|       | gain over node2vec | 28.7337 | 31.5043 | 43.1257 | 69.5343 |
|       | gain over SiNE   | 7.8984  | 10.5445 | 2.8296 | 0.1250 |

4. Performance measures (across all scores and across all algorithms) are comparatively better for Epinions over other datasets because almost 83% of the nodes in Epinions satisfy the structural balance condition \[11\]. As a result edge label prediction is comparatively easier than in other datasets.

5. The feature scoring method has a noticeable impact w.r.t. different datasets.
Table 4: Comparison of methods for node label prediction. SIGNet outperforms SiNE and node2vec in all datasets.

| Performance measure | Algorithms | ADJNet | SCOTUS |
|---------------------|------------|--------|--------|
| micro f1            | node2vec   | 0.5284 | 0.5392 |
|                     | SiNE       | 0.6257 | 0.6131 |
|                     | SIGNet-NS  | 0.7292 | 0.8004 |
|                     | SIGNet     | **0.8380** | **0.8419** |
| imp (%) over node2vec |            | 58.5920 | 56.1387 |
| imp (%) over SiNE   |            | 33.9300 | 37.3185 |
| imp (%) over SIGNet-NS |        | 14.9205 | 5.1849 |
| macro f1            | node2vec   | 0.4605 | 0.4922 |
|                     | SiNE       | 0.5847 | 0.5696 |
|                     | SIGNet-NS  | 0.7261 | 0.7997 |
|                     | SIGNet     | **0.8374** | **0.8415** |
| imp (%) over node2vec |            | 81.8458 | 70.9671 |
| imp (%) over SiNE   |            | 43.2187 | 47.7353 |
| imp (%) over SIGNet-NS |        | 15.3285 | 5.2270 |

The Average and Concatenation methods subsidize differences whereas the Hadamard, $L^1$ and $L^2$ methods promote differences. To understand why this makes a difference, consider networks like ADJNet and SCOTUS where connected components denote strong polarities (e.g., denoting synonyms or justice leanings, respectively). In such networks, the Hadamard, $L^1$ and $L^2$ methods provide more discriminatory features. On the other hand, Epinions and Slashdot are relatively large datasets with diversified communities and so all these methods perform nearly comparably.

### 4.5 Node Label Prediction

For datasets like SCOTUS and ADJNet (where nodes are annotated with labels), we learn a logistic regression classifier to map from node representations to corresponding labels (with a 50-50 training-test split). See Table 4 for results. As can be seen, SIGNet consistently outperforms the SiNE and node2vec approaches. In particular, in the case of SCOTUS which is a citation network, some cases have a huge number of citations (i.e., landmark cases) in both ideologies. Targeted node sampling, by adding such cases to either $\eta^+_i$ or $\eta^-_i$, situates the embedding space close to the landmark cases if they are in $\eta^+_i$ or away from them if they are in $\eta^-_i$, thus supporting accurate node prediction.

The case of Citizens United vs. Federal Election Commission (FEC), one of the most controversial cases in recent times, is instructive. In this case, Citizens United seeks an injection against the FEC to prevent the application of the Bipartisan Campaign Reform Act (BCRA) so that a film on Hillary Clinton can be broadcasted. In a 5-4 vote, the court decides in favor of Citizens United. In Fig. 7, we depict the BCRA related cases that cite Citizens United vs. Federal Election Commission in Fig. 7 (in a 2D projection). The cases whose decisions support a conservative view are shown in red and the cases which support a liberal point of view are shown in blue. Another two cases disputing the appli-
cation of BCRA cite this case (shown in filled circles), viz. Williams-Yulee vs The Florida Bar and McCutcheon vs FEC. In the first case the court supports the liberal point-of-view (shown in blue) and cites the case negatively (shown in dashed line). Therefore, its embedding resides far away from the Citizens United case. In McCutcheon vs FEC, the court supports a conservative point-of-view and decides in favor of McCutcheon. This case positively cites Citizens United case and its embedding is therefore positioned closer to it.

### 4.6 Node Label Prediction with Partial Information

To evaluate the effectiveness of our targeted node sampling versus negative sampling, we remove all outgoing edges of a certain percent of randomly selected nodes (test nodes), learn an embedding, and then aim to predict the labels of the test nodes. We show the F-1 scores for SCOTUS and ADJNet in Fig. 8. As seen here, SIGNet consistently outperforms SIGNet-NS. Withholding the outgoing edges of test nodes implies that both methods will miss the same edge information in learning the embedding. However due to targeted node sampling many of these test nodes will be added to \( \eta_i^+ \) or \( \eta_i^- \) in SIGNet (recall only the outgoing edges are removed, but not incoming edges). Because of this property, SIGNet will be able to make an informed choice while optimizing the embedding space.

### 4.7 Scalability

To assess the scalability of SIGNet, we learn embeddings for an Erdos-Renyi random network for up to one million nodes. The average degree for each node is set to 10 and the total number of samples is set to 100 times the number

![Figure 6: Predicting the polarity of adjectives in a subset of the ADJNet dataset. Here red labeled/boldface words are negative while the blue labeled/slanted words are positive. (Many adjectives have been removed to reduce clutter.) We use t-SNE to map the data into a 2D space.](image)
of edges in the network. The size of the dimension is also set to 100 for this experiment. The optimization time and the total execution time (targeted node sampling + optimization) is compared in Fig. 9 (a) for different vertex sizes. On a regular desktop, an unparallelized version of SIGNet requires less than 3 hours to learn the embedding space for over 1 million nodes. Moreover, the sampling times is negligible compared to the optimization time (less than 15 minutes for 1 million nodes). This actually shows SIGNet is very scalable for real world networks. Since SIGNet uses an asynchronous stochastic gradient approach, it is trivially parallelizable and as Fig. 9 (b) shows, we can obtain a 3.5 fold improvement with just 5 threads, with diminishing returns beyond that point.

4.8 Parameter Sensitivity

Fig. 10 (a) and (b) show that while we see improvements in macro F1 score for node label prediction upto 100 million samples, further optimization beyond that point is not necessary as the performance gets saturated. A similar behavior is seen with size of the embedding dimension in Fig. 10 (c) and (d) (saturation after 1000).

5 Other Related Work

Work related to unsupervised feature learning for networks have been discussed in the introduction. These ideas follow the trend opened up originally by unsupervised feature learning in text. Skip-gram models proposed in [22, 23] learn a
Figure 8: Performance evaluation on ADJNet and SCOTUS datasets varying the percent of nodes used for training. The $x$-axis shows the percent of nodes whose information is used to learn the embedding. $y$-axis shows the micro F1 score (top) and macro F1 score (bottom) for each dataset. SIGNet outperforms SIGNet-NS in all cases.

Figure 9: Scalability of SIGNet on Erdos-Renyi random graphs. (a) execution time of SIGNet varying the number of nodes; (b) execution time of SIGNet varying the number of threads.
vector representation of words by optimizing a likelihood function. Skip-gram models are based on the principle that words in similar contexts generally have similar meanings \cite{mikolov2013distributed} and can be extended to learn feature representation for documents \cite{mikolov2013distributed}, parts of speech \cite{mikolov2013distributed}, items in collaborative filtering \cite{mikolov2013distributed}. Recently deep learning based models have been proposed for representation learning on graphs to perform the above mentioned prediction tasks \cite{kipf2016semi, gloveme, you2018signed}. Although these models provide high accuracy by optimizing several layers of non-linear transformations, they are computationally expensive and requires a significant amount of training time as opposed to our proposed method SIGNet.

6 Discussion

We have presented a scalable feature learning framework suitable for signed networks. Using a targeted node sampling for random walks, and leveraging structural balance theory, we have shown how the embedding space learned by SIGNet yields interpretable as well as effective representations. Future work is aimed at extensions to networks with a heterogeneity of node and edge tables and at introducing additional semantic information into the representation.
learning process.

References

[1] O. Barkan and N. Koenigstein. ITEM2VEC: Neural item embedding for collaborative filtering. In Workshop on MLSP, pages 1–6, 2016.

[2] M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In NIPS, pages 585–591. MIT Press, 2001.

[3] S. Bhagat, G. Cormode, and S. Muthukrishnan. Node Classification in Social Networks, pages 115–148. Springer US, 2011.

[4] D. Cartwright and F. Harary. Structural balance: a generalization of Heider’s theory, 1956.

[5] J. Davis. Clustering and structural balance in graphs. Human Relations, 20:181–187, 1967.

[6] J. Tang et al. LINE: Large-scale Information Network Embedding. In WWW, pages 1067–1077, 2015.

[7] K. Li et al. Lrbm: A restricted boltzmann machine based approach for representation learning on linked data. In ICDM, pages 300–309, 2014.

[8] S. Yan et al. Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. IEEE Trans. PAMI, 29(1):40–51, 2007.

[9] X. Li et al. A deep learning approach to link prediction in dynamic networks. In SDM, pages 289–297, 2014.

[10] X. Yu et al. Personalized Entity Recommendation: A Heterogeneous Information Network Approach. In WSDM, pages 283–292, 2014.

[11] G. Facchetti, G. Iacono, and C. Altafini. Computing global structural balance in large-scale signed social networks. PNAS, 108(52):20953–20958, 2011.

[12] J. Fowler and S. Jeon. The authority of supreme court precedent. Social Networks, 30(1):16–30, 2008.

[13] A. Grover and J. Leskovec. node2vec: Scalable Feature Learning for Networks. In KDD, pages 855–864, 2016.

[14] P. Hage and F. Harary. Structural models in anthropology. Cambridge University Press, 1983.

[15] Z. Harris. Distributional Structure, pages 3–22. Springer Netherlands, 1981.

[16] F. Heider. Attitudes and Cognitive Organization. Journal of Psychology, 21:107–112, 1946.

[17] J. Kunegis, S. Stephan, A. Lommatzsch, J. Lerner, E. De Luca, and S. Albayrak. Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization. In SDM, pages 559–570, 2010.
[18] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In ICML, pages 1188–1196, 2014.

[19] A. Li, A. Ahmed, S. Ravi, and A. Smola. Reducing the sampling complexity of topic models. In KDD, pages 891–900, 2014.

[20] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. In ICLR, 2016.

[21] D. Liben-Nowell and J. Kleinberg. The Link Prediction Problem for Social Networks. In CIKM, pages 556–559, 2003.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. CoRR, abs/1301.3781, 2013.

[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In NIPS, pages 3111–3119. 2013.

[24] B. Perozzi, R. Al-Rfou, and S. Skiena. DeepWalk: Online Learning of Social Representations. In KDD, pages 701–710, 2014.

[25] K. Read. Cultures of the central highlands, new guinea. Southwest J Anthropol, 10(1):1–43, 1954.

[26] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In NIPS, pages 693–701, 2011.

[27] S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science, 290(5500):2323–2326, 2000.

[28] J. Tenenbaum, V. Silva, and J. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science, 290(5500):2319–2323, 2000.

[29] A. Trask, P. Michalak, and J. Liu. sense2vec-A fast and accurate method for word sense disambiguation in neural word embeddings. arXiv:1511.06388, 2015.

[30] L. van der Maaten and G. Hinton. Visualizing High-Dimensional Data Using t-SNE. JMLR, 9:2579–2605, 2008.

[31] S. Vishwanathan, N. Schraudolph, R. Kondor, and K. Borgwardt. Graph Kernels. JMLR, 11:1201–1242, 2010.

[32] S. Wang, J. Tang, C. Aggarwal, Y. Chang, and H. Liu. Signed network embedding in social media. In SDM, 2017.

[33] Q. Zheng and D. Skillicorn. Spectral Embedding of Signed Networks. In SDM, pages 55–63, 2015.