SIGNet: Scalable Embeddings for Signed Networks

Mohammad Raihanul Islam, B. Aditya Prakash, and Naren Ramakrishnan

Department of Computer Science, Virginia Tech., USA
{raihan8,badityap,naren}@cs.vt.edu

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. Such network embedding 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 and adds 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 [2], prediction [11], and visualization [12].
Classical approaches to find features and embeddings are motivated by dimensionality reduction research and extensions, e.g., approaches such as Laplacian eigenmaps [1], non-linear dimension reduction [17], and spectral embedding [7]. More recent research has focused on developing network analogs to distributed vector representations such as word2vec [13, 14]. In particular, by viewing sequences of nodes encountered on random walks as documents, methods such as DeepWalk [15], node2vec [5], and LINE [16] 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.
An attempt to fill this gap is the work of Wang et al. [19] 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 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 traditional negative sampling. The 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 technique can be treated as an extension of the negative sampling strategy 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 and node label prediction. SIGNet[1] is able to generate embeddings for networks with millions of nodes in a scalable manner.

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, and 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. $(v_i, v_j) \equiv (v_j, v_i)$). On the other hand, if G is directed then order becomes relevant (i.e. $(v_i, v_j) \not\equiv (v_j, v_i)$ 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 [5]) cannot be applied for signed networks. Consider a network wherein the nodes v_i and v_j are positively connected and the nodes v_k and v_i are negatively connected (see Fig. 1(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 v_i and v_k closer than v_i and v_j owing to the stronger influence of the weight (Fig. 1(b)). This problem remains unresolved even if we consider the weight of a negative edge as zero, because even though it may place node v_i and v_j closer, node v_k

[1] The implementation is available at: `https://github.com/raihan2108/signet`

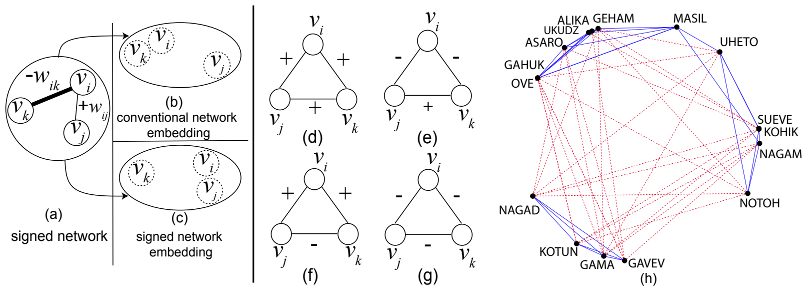


Fig. 1. *Given a signed network (a), a conventional network embedding (b) does not take signs into account and can result in faulty representations. (c) SIGNet learns embeddings that respect sign information between edges. Of the possible signed triangles, (d) and (e) are considered balanced but (f) and (g) are not. (h) shows a 2-dimensional embedding of alliances among New Guinea tribes using SIGNet. Alliance (hostility) between the tribes is shown in solid blue (dashed red) edges. We can see that edges representing alliances are comparatively shorter than the edges representing hostility.*

may be relatively closer to v_i because we ignore the adverse relation between node v_i and v_k. This may comprise the quality of embedding space. Ideally, we would like a representation wherein nodes v_i and v_j are closer than nodes v_i and v_k, as shown in Fig. 1(c). This example shows that modeling the polarity is as important as modeling the strength of the relationship.

To accurately model the interplay between the vertices in signed networks we use the theory of *structural balance* proposed in [6]. This theory posits that triangles with an odd number of positive edges are more plausible than an even number of positive edges (see Fig. 1 (d–g)). Although different adaptations of and alternatives to balance theory exist in the literature, here we focus on the original notion of structural balance to create the embedding space since it applies naturally to the experimental contexts considered here (e.g., networks constructed from adjectives, described in Sec. 4).

Problem Statement: *Scalable Embedding of Signed Networks (SIGNet):* Given a signed network G, compute a low-dimensional vector $\mathbf{d}_i \in \mathbb{R}^K$, $\forall v_i \in V$, where positively related vertices reside in close proximity to each other and negatively related vertices are distant from each other.

To explain the interpretability of the signed network embedding we utilize a small dataset denoting relations between 16 tribes in New Guinea. This is a signed network depicting alliances and hostility between the tribes. We learned the embeddings using SIGNet in 2 dimensional space as an undirected network as shown in Fig. 1(h). 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. Therefore the embedding space learned by SIGNet clearly depicts alliances and relationships among the tribes as intended.

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 $\mathbf{x}_i \in \mathbb{R}^K$. Then a natural way to compute the proximity between v_i and v_j is by the following function (ignoring the sign for now):

$$p_u(v_i, v_j) = \sigma(\mathbf{x}_j^T \cdot \mathbf{x}_i) = \frac{1}{1 + \exp(-\mathbf{x}_j^T \cdot \mathbf{x}_i)} \tag{1}$$

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}. $r_{ij} \in \mathbb{N}$ 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}, $p_u(v_i, v_j) = \sigma(s_{ij}(\mathbf{x}_j^T \cdot \mathbf{x}_i))$. Now incorporating the weight information, the objective function for undirected signed network can be written as:

$$\mathcal{O}_{un} = \sum_{e_{ij} \in E} r_{ij} \sigma(s_{ij}(\mathbf{x}_j^T \cdot \mathbf{x}_i)) = \sum_{e_{ij} \in E} r_{ij} p_u(v_i, v_j) \tag{2}$$

By maximizing Eqn. 2 we obtain a vector $\mathbf{x}_i$ of dimension K for each node $v_i \in V$ (we also use $\mathbf{d}_i$ to refer to 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 v_i and v_j should be proximal or not. We solve this problem by treating 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 v_j, node v_i should be closer. This enables us to treat all nodes consistently without worrying about reciprocity relationships. To this end, we introduce another vector $\mathbf{y}_i \in \mathbb{R}^K$ besides $\mathbf{x}_i$, $\forall v_i \in V$. For a directed edge e_{ij} the probability of context v_j given v_i is:

$$p_d(v_j | v_i) = \frac{\exp(s_{ij}(\mathbf{y}_j^T \cdot \mathbf{x}_i))}{\sum_{k=1}^{|V|} \exp(s_{ik}(\mathbf{y}_k^T \cdot \mathbf{x}_i))} \tag{3}$$

Treating the same entity as itself and as a specific context is very popular in the text representation literature [13]. 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:

$$\mathcal{O}_{dir} = \sum_{e_{ij} \in E} r_{ij} p_d(v_j | v_i) \tag{4}$$

By maximizing Eqn. 4 we will obtain two vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ for each $v_i \in V$. The vector $\mathbf{x}_i$ models the outward connection of a node whereas $\mathbf{y}_i$ models the inward connection of the node. Therefore the concatenation of $\mathbf{x}_i$ and $\mathbf{y}_i$ represents the final embedding for each node. We denote the final embedding of node v_i as $\mathbf{d}_i$. It should be noted that for undirected network $\mathbf{d}_i = \mathbf{x}_i$ whereas for a directed network $\mathbf{d}_i$ is the concatenation of $\mathbf{x}_i$ and $\mathbf{y}_i$. This means $|\mathbf{x}_i| = |\mathbf{y}_i| = \frac{K}{2}$ in the case of directed graph (for the same representational length).

3.3 Efficient Optimization by Targeted Node Sampling

The denominator of Eqn. 3 is very hard to compute as this requires marginalizing the conditional probability over the entire vertex set V. We adopt the classical negative sampling approach [14] wherein negative examples are selected from some distribution for each edge e_{ij}. However, for signed networks, conventional negative sampling does not work. For example consider the network from Fig. 2(a). Viewing this example as an unsigned network, while optimizing for edge e_{ij}, we will consider v_i and v_y as negative examples and thus they will be placed distantly from each other. However, in a signed network context, v_i and v_y have a friendlier relationship (than with, say, v_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.

Constructing the cache for each node: We aim to construct a cache of positive and negative examples for each node v_i where the positive (negative) example cache η_i^+ (η_i^-) contains nodes which should have a positive (negative) relationship with v_i according to structural balance theory. To construct these caches for each node v_i, we apply random walks of length l starting with v_i to obtain a sequence of nodes. Suppose the sequence is $\Omega =< v_i, v_{n_0}, \cdots, v_{n_{l-1}} >$. Now we add each node v_{n_p} to either η_i^+ or η_i^- by observing the estimated sign between v_i and v_{n_p}. The estimated sign is computed using the following recursive formula $\tilde{s}_{in_p} = \tilde{s}_{in_{p-1}} \times s_{n_{p-1}n_p}$. Here $\tilde{s}_{in_{p-1}}$ is the estimated sign between node v_i and node $v_{n_{p-1}}$, which can be computed recursively. The base case for this formula is $\tilde{s}_{in_1} = s_{in_0} \times s_{n_0n_1}$. If node v_{n_p} is not a neighbor of node v_i and $\tilde{s}_{in_p}$ is positive then we add v_{n_p} to η_i^+. On the other hand if $\tilde{s}_{in_p}$ is negative and v_{n_p} is not a neighbor of v_i then we add it to η_i^-. For example for the graph shown in Fig. 2(a), suppose a random walk starting with node v_i is $< v_i, v_j, v_k, v_z >$. Here node v_k will be added to η_i^+ as $\tilde{s}_{ik} = s_{ij} \times s_{jk} > 0$ (base case) and v_k is not a neighbor of v_i. However, v_z will be added to node η_i^- as $\tilde{s}_{iz} = \tilde{s}_{ik} \times s_{kz} < 0$ and v_z is not a neighbor of v_i.

The one problem with this approach is that a node v_j may be added to both η_i^+ and η_i^-. We denote this phenomena as *conflict* and define the reason for this *conflict* in Theorem 1. We resolve this situation by computing the shortest path between v_i and v_j and compute $\tilde{s}_{ij}$ between them using the shortest path,

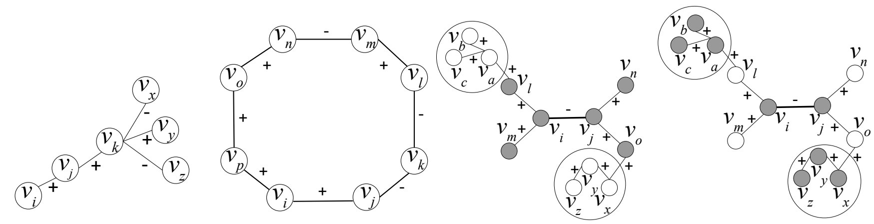


Fig. 2. *(a) depicts a small network to illustrate why conventional negative sampling does not work. v_i and v_y might be considered too distant for their representations to be placed close to each other. Targeted node sampling solves this problem by constructing a cache of nodes which can be used as sampling. (b) shows how we resolve conflict. Although there are two ways to proceed from node v_i to v_l the shortest path is v_i, v_j, v_k, v_l, which estimates a net positive relation between v_i and v_l. As a result v_l will be added to η_i^+. However for node v_m there are two shortest paths from v_i, with the path v_i, v_p, v_o, v_n, v_m having more positive edges but with a net negative relation, so v_m will be added to η_i^- in case of a conflict. (c) and (d) shows a comparative scenario depicting the optimization process inherent in both SiNE and SIGNet. The shaded vertices represent the nodes both methods will consider while optimizing the edge e_{ij}. We can see that 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 ($v_a, v_b, v_c, v_x, v_y, v_z$) for targeted node sampling.*

then add to either η_i^+ or η_i^- based on $\tilde{s}_{ij}$. 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 in case of a *conflict*, then we pick the path with the highest number of positive edges to compute $\tilde{s}_{ij}$. A scenario is shown in Fig. 2(b).

Theorem 1. *(Reason for conflict): Node v_j will be added to both η_i^+ and η_i^- if there are multiple paths from v_i to v_j and the union of these paths has at least one unbalanced cycle.*

Targeted edge sampling during optimization: Now after constructing the cache $\eta_i = \eta_i^+ \bigcup \eta_i^-$ for each node v_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 [14]. 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 (negative) edge e_{ij}, we aim to sample multiple negative (positive) nodes from η_i^- (η_i^+). Therefore the objective function for each edge becomes (taking log):

$$\mathcal{O}_{ij} = \log[\sigma(s_{ij}(\mathbf{y}_j^T \cdot \mathbf{x}_i))] + \sum_{c=1}^{\mathcal{N}} E_{v_n \sim \tau(s_{ij})} \log[\sigma(\tilde{s}_{in}(\mathbf{y}_n^T \cdot \mathbf{x}_i))] \tag{5}$$

Here $\mathcal{N}$ is the number of targeted node examples per edge and τ is a function which selects from η_i^+ or η_i^- based on the sign s_{ij}. τ selects from η_i^+ (η_i^-) if $s_{ij} < 0$ ($s_{ij} > 0$).

The benefit of targeted node sampling in terms of global balance considerations across the entire network is shown in Fig. 2 (c) and (d). Here we compare how our proposed approach SIGNet and SiNE [19] maintain structural balance. For simplicity suppose only edge e_{ij} has negative sign. Now SiNE 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 v_i and v_j, i.e. v_l, v_m, v_n, v_o (Fig. 2 (c)). However SIGNet skips the immediate neighbors while it uses higher order neighbors (i.e., $v_a, v_b, v_c, v_x, v_y, v_z$). Note that SIGNet actually uses immediate neighbors as separate examples (i.e edge e_{il}, e_{im} etc.). In this way SIGNet covers more nodes to optimize the embedding space than SiNE.

4 Experiments

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

– SNE [20]: This method computes the embedding using a log bilinear model; however it does not exploit any specific theory of signed networks.

– SiNE [19]: 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 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.

We skip hand crafted feature generation method for link prediction like [9] because they can not be applied in node label prediction and already shows inferior performance compared to SiNE. For node2vec the weight of negative edges are treated zero since node2vec can not handle negative edges.

In the discussion below, we focus on five real world signed network datasets. Out of these five, 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. The third dataset is a voting records of Wikipedia adminship election (Wiki), also from SNAP. The fourth dataset we study is an adjective network (ADJNet) constructed from the synonyms and antonyms collected from Wordnet database. Label information about whether the adjective is positive or negative comes from SentiWordNet. The last dataset is a citation network we constructed from

Table 1. *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	Wiki	SCOTUS	ADJNet
positive	0.86 (0.37)	0.98 (0.31)	1.06 (0.27)	0.84 (0.25)	0.71 (0.16)
negative	1.64 (0.23)	1.60 (0.19)	1.56 (0.19)	1.64 (0.21)	1.77 (0.08)
ratio	0.524	0.613	0.679	0.512	0.401

written case opinions of the Supreme Court of the United States (SCOTUS). We expand the notion of SCOTUS citation network into a signed network.
Unless otherwise stated, for directed networks we set $|\mathbf{x}_i| = |\mathbf{y}_i| = \frac{K}{2} = 20$ for both SIGNet-NS and SIGNet; therefore $|\mathbf{d}_i| = 40$. For a fair comparison, the final embedding dimension for others methods is set to 40. For undirected network (ADJNet) $|\mathbf{d}_i| = 40$ for all the methods. We also set the total number of samples (examples) to 100 million, $\mathcal{N} = 5$, $l = 50$ and $r = 1$ for SIGNet-NS and SIGNet. For all the other parameters for node2vec, SNE and SiNE we use the settings recommended in their respective papers.

Does the embedding space learned by SIGNet support structural balance theory? 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 1). 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 67% over all the datasets suggesting that SIGNet grasps the principles very effectively.

Are representations learned by SIGNet effective at edge label prediction? We now explore the utility of SIGNet for edge label prediction. For all the datasets we sample 50% 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 50% of the edges. Since edges involve two nodes we explore several scores to compute the features for edges from the node embedding. They are **Concatenation:** (concat): $\mathbf{f}_{ij=}\mathbf{d}_i \oplus \mathbf{d}_j$, **Average** (avg): $\mathbf{f}_{ij=}\frac{\mathbf{d}_i+\mathbf{d}_j}{2}$, **Hadamard** (had): $\mathbf{f}_{ij=}\mathbf{d}_i * \mathbf{d}_j$, $\mathcal{L}_1$: $\mathbf{f}_{ij=}|\mathbf{d}_i - \mathbf{d}_j|$ and $\mathcal{L}_2$: $\mathbf{f}_{ij=}|\mathbf{d}_i - \mathbf{d}_j|^2$.
Here $\mathbf{f}_{ij}$ is the feature vector of edge e_{ij} and $\mathbf{d}_i$ is the embedding of node v_i. Except for the method of concatenation (which has a feature vector dimension of 80) other methods use 40-dimensional vectors. We use the micro-F1 scores to

Table 2. *Comparison of edge label prediction in all datasets. We show the micro F1 score for each feature scoring method. The best score across all the scoring method is shown in boldface. SIGNet outperforms node2vec, SNE, and SiNE in every case. The results are statistically significant with* $p < 0.01$.

Eval.	Dataset Name	Epinions	Slashdot	Wiki	ADJNet	SCOTUS
concat	node2vec	0.831	0.776	0.749	0.594	0.543
	SNE	0.854	0.778	0.751	0.602	0.528
	SiNE	0.856	0.779	0.752	0.598	0.605
	SIGNet-NS	0.911	0.793	0.816	0.599	0.56
	SIGNet	**0.920**	**0.832**	**0.845**	0.573	0.557
avg	node2vec	0.853	0.775	0.747	0.603	0.516
	SNE	0.853	0.776	0.748	0.601	0.532
	SiNE	0.853	0.774	0.749	0.599	0.608
	SIGNet-NS	0.837	0.771	0.769	0.620	0.509
	SIGNet	0.879	0.809	0.801	0.574	0.512
had	node2vec	0.852	0.773	0.748	0.600	0.562
	SNE	0.851	0.775	0.745	0.604	0.541
	SiNE	0.854	0.772	0.748	0.589	0.609
	SIGNet-NS	0.846	0.757	0.741	0.705	**0.793**
	SIGNet	0.883	0.782	0.754	**0.722**	0.792
l1	node2vec	0.852	0.775	0.747	0.601	0.559
	SNE	0.854	0.774	0.749	0.605	0.582
	SiNE	0.853	0.773	0.746	0.609	0.608
	SIGNet-NS	0.851	0.764	0.743	0.639	0.723
	SIGNet	0.901	0.787	0.751	0.703	0.723
l2	node2vec	0.852	0.773	0.747	0.601	0.569
	SNE	0.852	0.774	0.748	0.606	0.547
	SiNE	0.787	0.776	0.745	0.612	0.611
	SIGNet-NS	0.848	0.763	0.743	0.659	0.742
	SIGNet	0.903	0.809	0.753	0.716	0.745
gain over node2vec		7.85	7.22	12.82	19.73	39.19
gain over SNE		7.73	6.94	12.52	19.14	36.08
gain over SiNE		7.48	6.80	12.37	17.97	29.62
gain over SIGNet-NS		0.99	4.92	3.55	2.41	-0.13

evaluate our method. We repeat this process five times and report the average results (see Table 2). 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 Wiki the gains are moderate (around 12%) where 25% of edges are negative. For ADJNet and SCOTUS where the sign distribution is less skewed, SIGNet outperforms node2vec by a huge margin (19% for ADJNet and 39% for SCOTUS).
2. SIGNet demonstrates a consistent advantage over SiNE and SNE, with gains ranging from 6–12% (for the social network datasets) to 17–36% (for ADJNet and SCOTUS).
3. SIGNet also outperforms SIGNet-NS in almost all scenarios demonstrating the effectiveness of targeted node sampling over negative sampling.
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 [3]. As a result in Epinions edge label prediction is comparatively easier than in other datasets.
5. The feature scoring method has a noticeable impact w.r.t. different datasets. The avg. and concat. methods subsidize differences whereas the hadamard, $\mathcal{L}$-1

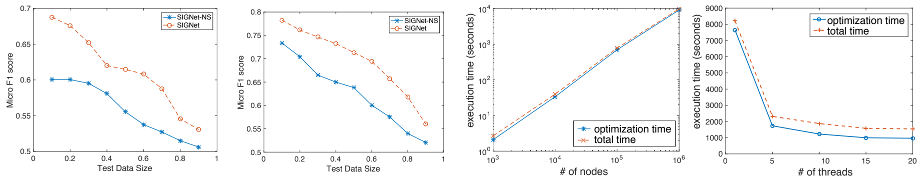
Table 3. *Comparison of methods for node label prediction on real world datasets. SIGNet outperforms other methods in all datasets.*

Metric	micro f1		macro f1	
Datasets	ADjNet	SCOTUS	ADjNet	SCOTUS
node2vec	0.5284	0.5392	0.4605	0.4922
SNE	0.5480	0.5432	0.4840	0.5335
SiNE	0.6257	0.6131	0.6247	0.5796
SIGNet-NS	0.7292	0.8004	0.7261	0.7997
SIGNet	0.8380	0.8419	0.8374	0.8415
gain over node2vec	58.5920	56.1387	81.8458	70.9671
gain over SNE	52.9197	54.9890	73.0165	57.7320
gain over SiNE	33.9300	37.3185	34.0483	45.1863
gain over SIGNet-NS	14.9205	5.1849	15.3285	5.2270

and $\mathcal{L}$-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, $\mathcal{L}$-1 and $\mathcal{L}$-2 methods provide more discriminatory features. However, Epinions and Slashdot are relatively large datasets with diversified communities and so all these methods perform nearly comparably.

Are representations learned by SIGNet effective at 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). We also repeat this five times and report the average. See Table 3 for results. As can be seen, SIGNet consistently outperforms all the other 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 η_i^+ or η_i^-, situates the embedding space close to the landmark cases if they are in η_i^+ or away from them if they are in η_i^-, thus supporting accurate node prediction.

How much more effective is our sampling strategy in the presence of 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 micro f1 scores for ADJNet (treating it as directed) and SCOTUS in Fig. 3 (a) and (b). 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 η_i^+ or η_i^- in SIGNet (recall only the outgoing edges are removed, but not incoming edges). Because of this property, SIGNet is able to make an informed choice while optimizing the embedding space.



(a) (b) (c) (d)

Fig. 3. *Micro F1 of ADJNet (a) and SCOTUS (b) datasets varying the percent of nodes used for training. SIGNet outperforms SIGNet-NS in all cases. (c) and (d) show execution time of SIGNet varying the number of nodes and threads.*

How scalable is SIGNet for large networks? To assess the scalability of SIGNet, we learn embeddings for an Erdos-Renyi random network for upto 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 of edges in the network. The size of the dimension is also set to 100 for this experiment. We make the network signed by randomly changing the sign of 20% edges to negative. The optimization time and the total execution time (targeted node sampling + optimization) is compared in Fig. 3 (c) 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. Additionally, SIGNet uses an asynchronous stochastic gradient approach, so it is trivially parallelizable and as Fig. 3(d) shows, we can obtain a 3.5 fold improvement with just 5 threads, with diminishing returns beyond that point.

5 Other Related Work

The concept of unsupervised learning in networks follow the trend opened up originally by Skip-grapm models [13, 14]. Skip-gram models can be extended to learn feature representations for documents [8], diseases [4] etc. Recently deep learning based models have been proposed for representation learning on graphs to perform prediction in unsigned networks [10, 18]. Although these models provide high accuracy by optimizing several layers of non-linear transformations, they are computationally expensive, require a significant amount of training time and are only applicable to unsigned networks as opposed to our proposed method SIGNet.

6 Conclusion

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 experimenting with other theories of signed networks and extensions to networks with a heterogeneity of node and edge tables.

Acknowledgments: This research has been supported by US NSF grants IIS-1750407 (CAREER), IIS-1633363, and DGE-1545362, by the NEH (HG-229283-15), by ORNL (Task Order 4000143330), by the Maryland Procurement Office (H98230-14-C-0127), and by a Facebook faculty gift.

References

1. Belkin, M., Niyogi, P.: Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In: NIPS (2001)
2. Bhagat, S., Cormode, G., Muthukrishnan, S.: Node Classification in Social Networks. Springer US (2011)
3. Facchetti, G., Iacono, G., Altafini, C.: Computing global structural balance in large-scale signed social networks. PNAS 108(52) (2011)
4. Ghosh, S., et al.: Characterizing diseases from unstructured text: A vocabulary driven word2vec approach. In: CIKM (2016)
5. Grover, A., Leskovec, J.: node2vec: Scalable Feature Learning for Networks. In: KDD (2016)
6. Heider, F.: Attitudes and Cognitive Organization. Journal of Psychology 21 (1946)
7. Kunegis, J., et al.: Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization. In: SDM (2010)
8. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: ICML (2014)
9. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting Positive and Negative Links in Online Social Networks. In: WWW (2010)
10. Li, K., et al.: LRBM: A restricted boltzmann machine based approach for representation learning on linked data. In: ICDM (2014)
11. Liben-Nowell, D., Kleinberg, J.: The Link Prediction Problem for Social Networks. In: CIKM (2003)
12. van der Maaten, L., Hinton, G.: Visualizing High-Dimensional Data Using t-SNE. JMLR 9 (2008)
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR abs/1301.3781 (2013)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: NIPS (2013)
15. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: Online Learning of Social Representations. In: KDD (2014)
16. Tang, J., et al.: LINE: Large-scale Information Network Embedding. In: WWW (2015)
17. Tenenbaum, J., de Silva, V., Langford, J.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290(5500) (2000)
18. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: KDD (2016)
19. Wang, S., Tang, J., Aggarwal, C., Chang, Y., Liu, H.: Signed network embedding in social media. In: SDM (2017)
20. Yuan, S., Wu, X., Xiang, Y.: SNE: Signed network embedding. In: PAKDD (2017)