Structural Node Embedding in Signed Social Networks: Finding Online Misbehavior at Multiple Scales

Mark Heimann[1], Goran Murić[2], and Emilio Ferrara[2]

[1] Lawrence Livermore National Laboratory, Livermore, CA, USA**
heimann2@llnl.gov
[2] Information Sciences Institute, Marina Del Rey, CA, USA
gmuric@isi.edu, ferrarrae@isi.edu

Abstract. Entities in networks may interact positively as well as negatively with each other, which may be modeled by a signed network containing both positive and negative edges between nodes. Understanding *how* entities behave and not just with *whom* they interact positively or negatively leads us to the new problem of structural role mining in signed networks. We solve this problem by developing structural node embedding methods that build on sociological theory and technical advances developed specifically for signed networks. With our methods, we can not only perform node-level role analysis, but also solve another new problem of characterizing entire signed networks to make network-level predictions. We motivate our work with an application to social media analysis, where we show that our methods are more insightful and effective at detecting user-level and session-level malicious online behavior from the network structure than previous approaches based on feature engineering.

1 Introduction

Networks are a natural model for many forms of data, in which entities exhibit complex patterns of interaction. In many applications, entities may form negative as well as positive interactions with each other. For example, users on a social network may form friendships and engage in prosocial behavior with each other, but they may also form animositities and engage in antisocial behavior, such as trolling [17] or cyberaggression [10]. *Signed networks*, in which edges between node may be positive or negative, can naturally model these interactions of varying polarity.

Existing work in signed network analysis often tries to characterize *with whom* each node interacts. For example, the common task of *edge sign prediction* [1] is to determine whether two nodes would have a positive or negative interaction; *node embedding* objectives for signed networks [22, 2] encourage each node to have similar latent feature representations to other nodes with whom it interacts positively, and dissimilar representations to those with whom it interacts negatively. Instead, we focus on the orthogonal and new problem of characterizing *how* a node forms positive *or negative* relationships, namely its *structural role* in the signed network. Existing methods for role analysis in networks [19] are designed for unsigned networks, so we introduce

** Work done while author was an intern at the Information Sciences Institute.

structural node embedding methods that build on sociological theories and technical advances designed specifically for signed networks.

Not only can we perform node-level structural role analysis, but we can also characterize network-level behavior. For signed networks, this is another new problem, as methods for graph comparison and classification focus on unsigned networks. To solve it, we contribute baseline statistical signatures, graph kernels, and graph features derived from our signed structural node embeddings. The latter describe a signed network's *distribution* of structural roles, a very natural and powerful way to characterize a network.

We motivate our methodology with an application to social media analysis. Several types of social media users are revealed by their patterns of negative behavior on platforms designed to encourage positive interactions. For instance, *trolls* on social media may try to stir up controversy or causing annoyance to others [17], while *cyberbullies* leave hurtful or aggressive comments for other social media users with the intent to shame, mock, and/or intimidate [11]. Further understanding the nature of antisocial online behavior can lead to more effective preventative measures and improve the social media experience. Our node-level and graph-level methods respectively allow us to characterize social behavior at the level of individual users and larger media sessions. While recent work derived several insights about users' social roles from the network structure alone using hand-engineered network statistics [15], we show that our embedding-based methods can improve on network feature engineering.

Our contributions are thus as follows:

1. **New Node-level Problem and Methods**: we propose structural node embedding methods for the new problem of structural role analysis in signed networks. We propose a simple scheme to leveraging existing (unsigned) methods directly, and also new embedding methods that can learn from multi-sign higher-order interactions.
2. **New Graph-level Problem and Methods**: we propose techniques for the new problem of signed network classification: baselines using hand-engineered features and more expressive methods that leverage our signed structural node embeddings.
3. **Controlled Synthetic Test Cases for Signed Networks:** We study the behavior of our techniques in synthetic networks, where we can tailor the signed structural roles of nodes. Our methods are more discriminative than the simpler feature engineering approaches that have recently been used to characterize similar social roles.
4. **Applications to Social Media Analysis**: We use social media datasets with explicit user-specified edge signs and implicit signs that must be inferred, which can serve as benchmarks for our new problems. In node-level and graph-level analysis of social roles, our methods yield quantitative improvements over feature engineering.

For reproducibility, our code is available at `https://github.com/markheimann/Signed-Network-Roles`.

2 Related Work

The majority of works in graph mining focus on unsigned graphs. We first review relevant literature from the unsigned graphs, noting methods designed for signed networks usually outperform unsigned methods naïvely applied to signed networks [2]. With this in mind, we also review relevant literature from signed networks.

2.1 Unsigned Network Methods

We first review node embedding methods for unsigned networks, which enables node-level graph mining; we then discuss graph-level analysis.

Node Embedding. Node embedding learns features for nodes in a network via an objective that encourages similar nodes to have similar features. Often, similarity is defined based on node proximity. Unsupervised methods may model higher order proximities using random walks, matrix factorization, or deep neural networks; we refer the reader to a comprehensive survey on (proximity-preserving) node embedding [6] for more details. In a semi-supervised setting, graph neural networks [16] have grown in popularity. A complementary line of work embeds nodes based on their structural roles: nodes will be embedded close to other nodes with similar local structure, regardless of proximity. Such *structural embedding* methods are surveyed and contrasted conceptually [20] and empirically [14] to proximity-preserving embedding methods.

Graph Classification. While the above methods are often used for node-level analysis, many applications also require network-level predictions. Graph classification seeks to predict the class to which an entire network belongs using supervised machine learning. Three major families of techniques include kernels operating on graph similarity functions [21], unsupervised graph feature learning [8], and end-to-end feature learning with deep neural networks on graphs [5]. Given node embeddings that are comparable across networks–in particular, embeddings capturing structural roles–the gap between node-level and graph-level features can be bridged by modeling the distribution of node embeddings in a sparse graph feature vector. [8].

2.2 Signed Network Methods

Many of the works involved in analyzing signed social networks have their underpinnings sociological theory. One of the most influential theories on signed network mining is *balance theory* [7], which specifies some general rules for "balanced" and "unbalanced" configurations of edge signs. It predicts that balanced configurations are more stable and thus more likely to occur than unbalanced configurations, and generalizes intuition often captured in expressions such as "my enemy's enemy is my friend".

One of the principal tasks in signed network mining is edge sign prediction. This may be done by directly trying to minimize imbalance in the network, construction of hand-engineered signed features for supervised prediction, or matrix factorization [1]. Recent works have extended network embedding to signed networks based on shallow architectures and random walks [23] or deep architectures [22], applying them to node-level and edge-level tasks within a single network (to the best of our knowledge, multi-network tasks on signed social networks are largely unexplored). Existing approaches do not model structural roles, but rather preserve signed proximity in the network, trying to learn similar embeddings for positively connected nodes and different embeddings for negatively connected nodes.

3 Preliminaries

Let $G = (V, E)$ be a directed graph with vertex set V ($|V| = n$) and edge set $E \subseteq V \times V$. In this work, we consider *signed* networks: a sign function $\varphi : E \to \{1, -1\}$

dictates the sign of each edge as being positive or negative. G has single-sign subgraphs G^+ induced by the positive edges of G and G^- induced by the negative edges. G has an adjacency matrix $\mathbf{A}$, whose nonzero entries are ± 1 and whose i-th row is $\mathbf{A}_i$.

The goal of *node embedding* is to learn low-dimensional features for each node that capture higher-order network structure. Nodes are embedded into d-dimensional space, where d is a small constant, such that their geometry in vector space preserves some sort of node similarity in the original network (in our case, structural role similarity). $\mathbf{Y} \in \mathbb{R}^{n \times d}$ denotes a graph's matrix of node embeddings.

4 Node-Level Techniques

In this section, we extend structural node embedding [9, 4] to signed networks. We first discuss how existing unsigned network embedding methods can be applied directly on signed networks (§4.1) and the strengths and weaknesses of doing so. To overcome limitations of this approach, we adapt two unsigned structural embedding methods to the signed network domain (§4.2 & 4.3) using principled techniques for modeling signed network structure more expressively.

4.1 sec-Embedding: Concatenation of Single-Sign Embeddings

To apply unsigned node embedding methods directly to signed networks, we can use them to model separate structural roles of roles based only on positive or negative edges. Formally, given an embedding method, we apply it to G^+ to produce an embedding $\mathbf{Y}^+$, as well as to G^- to produce an embedding $\mathbf{Y}^-$. The final embedding is $\mathbf{Y} = \mathbf{Y}^+||\mathbf{Y}^-$. We call this heuristic *single-signed embedding concatenation*, which we denote by appending the prefix sec- before an embedding name (e.g. sec-xNetMF).

The sec– technique is simple, allowing unsigned structural embedding methods to be applied to signed networks without methodological modification; moreover, it may offer benefits of interpretability and generalizability by disentangling the effect of positive and negative edges on the structural roles. However, it ignores the complex structure of higher-order mixed-sign interactions present in signed networks, which have been characterized by rich sociological theory [7] that has informed signed network mining [1]. Discovery of some meaningful structural role similarities and differences, as we show concretely in §6.1, requires methodological innovation to capture structural roles based on higher-order mixed-sign structure.

4.2 sNCE: Embedding Nodes based on Signed Neighborhood Connectivities

The first structural node embedding method we adapt to signed networks is xNetMF [9], which embeds unsigned, undirected graphs from an implicit decomposition of a pairwise structural node similarity matrix. Nodes' structural similarity depends on the distribution of structural features (degree) in their k-hop neighborhoods; EMBER [13] generalizes this to handle edge directions by modeling separate neighborhoods based on incoming and outgoing edges. However, gracefully modeling edge signs poses two research challenges: $\mathrm{C}_{\mathbf{x}}1$) For each node, how do we **model higher-order neighbors'**

***signed* relationships** to that node? (We need to model positive and negative neighbors separately, and discern whether indirect neighbors are indirectly positive or negative.) $\mathbf{C_x}2$) How do we **model nodes' *signed* connectivity** in a neighborhood? (Degree alone, the usual connectivity measure [18, 9] does not incorporate edge signs.)

We call our proposed signed structural embedding method **sNCE** (signed Neighborhood Connectivity Embedding) as we apply the fundamental idea of xNetMF–embedding nodes based on structural similarity, derived from the connectivity statistics in their local neighborhoods–while respecting best practices on signed networks.

Defining Neighborhoods. To address $\mathbf{C_x}1$, we turn to sociological theories of *balance* [7]: we partition neighborhoods into balanced and unbalanced neighborhoods [2]. Formally, for a node u, let $\mathcal{N}_u^{k\rightarrow}$ be the k-hop out-neighborhood of u: the nodes that can be reached from u in a directed path of length exactly k. For immediate (one-hop) neighborhoods, balanced and unbalanced neighborhoods depend on the edge sign between the node and the neighbor: $\mathcal{B}_u^{1\rightarrow} = \{v \in \mathcal{N}_u^{k\rightarrow} : G(u,v) > 0\}$ and $\mathcal{U}_u^{1\rightarrow} = \{v \in \mathcal{N}_u^{k\rightarrow} : G(u,v) < 0\}$. For $k > 1$, we recursively define balanced higher-order neighborhoods [2]. The balanced k-hop neighborhood of u consists of all positive neighbors of balanced $(k-1)$-hop neighbors of u ("friends of friends"), as well as negative neighbors of unbalanced $(k-1)$-hop neighbors of u ("enemies of enemies"): $\mathcal{B}_u^{k\rightarrow} = \{v : v \in \mathcal{B}_{v'}^{(k-1)\rightarrow} : G(v',v) > 0\} \cup \{v : v \in \mathcal{U}_{v'}^{(k-1)\rightarrow} : G(v',v) < 0\}$. The unbalanced k-hop neighborhood of u consists of all negative neighbors of balanced $(k-1)$-hop neighbors of u ("enemies of friends"), as well as positive neighbors of unbalanced $(k-1)$-hop neighbors of u ("friends of enemies"): $\mathcal{U}_u^{k\rightarrow} = \{v : v \in \mathcal{U}_{v'}^{(k-1)\rightarrow} : G(v',v) > 0\} \cup \{v : v \in \mathcal{B}_{v'}^{(k-1)\rightarrow} : G(v',v) < 0\}$. Balanced and unbalanced in-neighborhoods are defined analogously using $\mathcal{N}_u^{k\leftarrow}$, the k-hop in-neighborhood of u or the nodes that can reach u via a directed path of length k.

Characterizing Neighborhoods. After splitting the k-hop neighborhoods of each node into balanced and unbalanced in- and out-neighborhoods following [2], we characterize the original node's structural role while respecting $\mathbf{C_x}2$ by examining several signed structural connectivity measures in these neighborhoods. Let $\mathcal{F}$ be the set of connectivity measures; $|\mathcal{F}| = 4$ as it consists of positive and negative node in- and out-degree. We consider each connectivity measure's distribution in the set N of neighborhoods of u; $|N| = 4$ as it consists of balanced and unbalanced in- and out-neighborhoods. Then, for $f \in \mathcal{F}$, the distribution of f in node u's (balanced/unbalanced, in/out) k-hop neighborhood $\mathcal{N}_u^k$ can be represented in a logarithmically binned [9] histogram $h_f(\mathcal{N}_u^k)$. We combine hop distances, discounting further ones [9, 13]: $h_f(\mathcal{N}_u) = \sum_{k=1}^{K} \delta_k h_f(\mathcal{N}_u^k)$, using maximum distance $K = 2$ [9, 13] and discount factor $\delta = 0.9$ by default. Nodes' structural similarity can be computed by comparing these histograms:

$$sim(u,v) = \exp(-\sum_{f\in\mathcal{F}} \sum_{\mathcal{N}\in N} ||h_f(\mathcal{N}_u) - h_f(\mathcal{N}_v)||) \tag{1}$$

Embedding. To learn structural node embeddings, we want nodes to have similar features if they are structurally similar according to Equation 1. Following the scalable implicit matrix factorization approach of [9], we derive embeddings from a low-rank decomposition of a pairwise node structural similarity matrix. To compute d-dimensional embeddings, we select d landmark nodes uniformly at random [9] and compute the $n \times d$

similarity matrix $\mathbf{C}$ of all nodes to these landmarks using Equation 1. We then form the $d \times d$ submatrix $\mathbf{W}$ of landmark-to-landmark similarities. With the pseudoinverse of the $\mathbf{W}$ and its SVD $\mathbf{W}^{\dagger} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$, we form the embeddings: $\mathbf{Y} = \mathbf{C}\mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}$.

4.3 sRDE: Embedding Nodes based on Distributions of Signed Relevance Scores

GraphWave [4] computes a matrix representing pairwise node relevance scores; each nodes' structural embedding models the *distribution* of its relevance to other nodes. While for unsigned networks, the relevance scores can be derived from heat diffusion, it is not clear how how this diffusion process would respect edge signs (we found that using it led to poor performance). Our research challenges include $\mathrm{C_g}1$: how can we **compute appropriate *signed* relevance scores**, and $\mathrm{C_g}2$: how do we appropriately **model the score *distributions***?

We call our proposed signed structural embedding method **sRDE** (signed Relevance Distribution Embedding) as we apply the fundamental idea of GraphWave–embedding distributions of relevance scores–while respecting best practices on signed networks.

Computing Node Relevance. To address $\mathrm{C_g}1$, we compute node relevance using *signed random walk with restart (RWR)* [3], which has a closed form matrix expression:

$$\mathbf{R} = (1-c)(\mathbf{I} - c\mathbf{S})^{-1} \tag{2}$$

where $c \in [0,1]$ is a scalar; $\overline{\mathbf{D}}$ is the signed degree matrix, a diagonal matrix where $\overline{\mathbf{D}}_{ii} = \sum_j |A_{ij}|$; $\mathbf{S} = \overline{\mathbf{D}}^{-1}\mathbf{A}$ is the signed random walk transition matrix. (In the future, iterative methods may be used to scale the computation of $\mathbf{R}$.)

Embedding. To address $\mathrm{C_g}2$, for a node u, we form the embedding $\mathbf{Y}_u$ by computing histogram over its relevance scores to all other nodes, given by the u-th row of the signed RWR matrix $\mathbf{R}_u$. Using d evenly spaced bins, we represent each node as a d-dimensional vector. We find this to be a simpler and empirically more effective alternative to sampling from the empirical characteristic function computed from $\mathbf{R}_u$, as proposed to learn structural embeddings from the (unsigned) heat kernel matrix [4].

5 Graph-Level Techniques

The techniques in §4 produce a single feature vector for each node, which may be used for node-level analysis. However, we may also want to analyze entire networks. Existing methods for graph comparison focus on unsigned networks, so we extend statistical and kernel-based methods to signed networks (§5.1). These approaches use hand-engineered features which may be less expressive than node embeddings. Thus, we leverage a recent unsupervised graph feature learning technique [8] to directly turn our node features from §4 into more expressive graph features for entire (signed) networks.

5.1 Signed Network Statistical Signatures and Kernels

As baselines, we propose methods for comparing graphs based on hand-engineered signed network statistics, using two different graph comparison methods: graph statistical signatures and graph kernels.

Statistical Signatures. We can construct feature vectors based on hand-engineered statistics in the graph. One simple approach is **Signed Maximum Degree (SMD)**: we form a four-dimensional feature vector consisting of the maximum positive and negative in- and out-degrees of any nodes in the graph. Such a feature vector captures some simple structural properties, but of course discards information about the degrees of most of the nodes. Thus, we consider the **Signed Degree Distribution (SDD)**: we form and concatenate histograms for the distribution of positive and negative in- and out-degrees in the networks. Each histogram has one bucket for each possible degree statistic, up to the maximum value of that statistic for any node in the entire dataset.

Graph kernel. We can compute a kernel on graphs based on the distribution of motifs in the graph [21]. Each graph G has a feature vector $\phi(G)$ counting the number of unique graphlets, or graph structures of consisting of k nodes, appear in the graph. (Usually k is a small number, with 3 being a popular choice in the literature). The graphlets kernel between two graphs is then given as the inner product of their feature vectors $k(G_1, G_2) = \langle \phi(G_1), \phi(G_2) \rangle$. Our **signed graphlet kernel (SGK)** counts the number of configurations of unique 3-node graphlets counting edge signs. That is $\phi(G)_i = \#$ of times the i-th signed graphlet appears in G. Using 1 to denote a positive edge, -1 to denote a negative edge, and 0 to denote no edge, we consider all unique combinations of 0s and ± 1. $\phi(G)$ is thus a vector with ten elements corresponding to counts of each of these graphlets, which we normalize to sum to one.

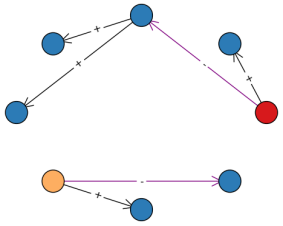
5.2 Signed RGM: Distributions of Signed Structural Node Embeddings

Given any set of node embeddings for a graph, the RGM feature map [8] represents a graph as a histogram of the distribution of its node embeddings in vector space. When these embeddings reflect structural roles, RGM models the **distribution of structural roles** in a network. With it, we turn any of our node embeddings from § 4 into graph features with a clear interpretation, which may be used for graph-level learning.

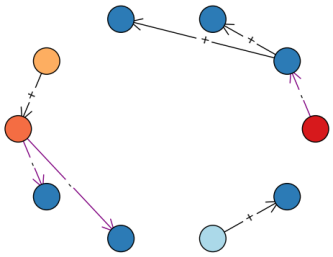
We follow all steps of the RGM procedure [8]: we normalize the embeddings and bin them using a partition of $[0, 1]^d$ given by a d-dimensional grid with cell widths μ and offsets δ sampled independently along each dimension: $\mu \sim \text{Gamma}(2, 1/\gamma)$, and $\delta \sim \text{unif}(0, \mu)$. The c-th entry of the histogram counts the number of node embeddings that fall into the c-th cell of the grid: these histograms form a sparse feature map for the graph. The parameter γ controls the resolution of the histograms, similar to a RBF or Laplacian kernel. To capture multiresolution structure, we concatenate histograms chosen by $\gamma \in [1, 2, 4, 8]$, weighted by $\frac{1}{\gamma}$ to place greater emphasis on matches found in tighter histograms; we also use two iterations of label expansion, starting with uniform node labels, to topologically group nodes prior to binning.

6 Experiments

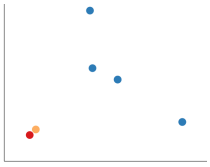
We first use controlled synthetic experiments to illustrate theoretical expressivity of various embedding methods, before using our node- and graph-level techniques to conduct real-world social media analysis.



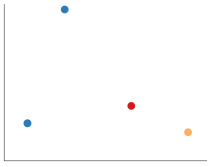
(a) Graph G_1 with role differences



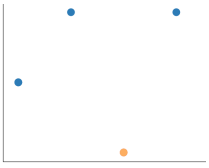
(b) Graph G_2 with role similarities



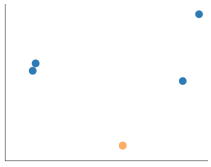
(c) G_1: sNCE finds differences



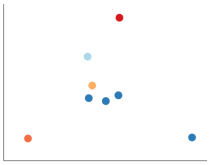
(d) G_1: sRDE finds differences



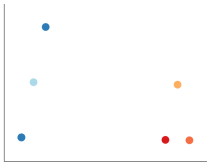
(e) G_1: Degree features do not find differences



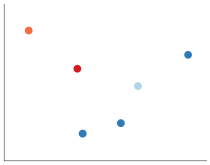
(f) G_1: sec-xNetMF does not find differences



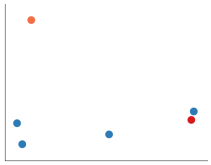
(g) G_2: sNCE finds differences but not similarities



(h) G_2: sRDE finds differences and similarities



(i) G_2: Degree features do not find similarities or some differences



(j) G_2: sec-xNetMF does not find similarities or some differences

Fig. 1: Synthetic graphs: Signed structural node embeddings can distinguish structurally different nodes (in G_1) and recognize structurally similar nodes (in G_2) using higher-order connections or information from multi-sign paths. Simpler approaches based on degree features or combining single-sign embeddings cannot do this.

6.1 Role Discovery in Synthetic Networks

To understand what our embeddings can learn in a controlled context, we generate signed networks with planted structural roles, a form of analysis often used for unsigned structural embeddings [14]. Our graphs contain disconnected components, but structural roles do not rely on node proximity and disconnected nodes can be compared [9]. We learn 4-dimensional node embeddings due to the small size of the graphs, and visualize the nodes' embedding similarity in two dimensions using PCA.

For the graph drawn in Fig. 1a, the red and yellow nodes have similar but not identical structural roles, when higher order connections are considered (in fact, the yellow

node has no 2-hop neighbors, but the red node does). **Degree statistics cannot capture higher order information**, so these nodes are given identical structural roles (Fig. 1e) and overlap in the plot; hence, the red node is not visible. Network embedding can model higher order information, but **embedding positive and negative components separately loses information from mixed-sign connections**. Concatenating unsigned node embeddings (§ 4.1) cannot distinguish between these nodes either (Fig. 1f), since the only higher-order neighborhoods have mixed signs. However, signed structural embedding methods can give these roles different embeddings.

For the graph shown in Fig. 1b, we highlight three nodes with 'warm' colors red, orange, and yellow, as they have similar structural roles analogous to some patterns of online (mis)behavior. A user behaving like the red or orange nodes, sending negative edges to nodes without additional negative edges, might be actively antagonizing ordinary users, while a user behaving like the yellow node might be goading an antagonizer on (sending positive edges to nodes that send negative edges): both propagate largely negative influence throughout the network [15].

Signed structural embeddings such as sNCE and sRDE capture this, embedding the two nodes similarly in the vector space. (Indeed, the entire goal of sRDE embeddings is to characterize the signed propagation patterns from each node). However, without higher-order, multi-sign connections, we cannot distinguish the behavior of goading on a bully (like the yellow node) from supporting a normal user (like the node marked in light blue). Thus, the yellow node is invisible, as it overlaps with the light blue node in Figures 1i and 1j which plot the features learned by concatenating single-signed embeddings or using hand-engineered statistics. sNCE (Figure 1g) can model these differences, but does not recognize the similarity of the yellow, red, and orange nodes. On the other hand, sRDE (Figure 1h) successfully clusters these together.

6.2 Finding Misbehavior in Social Media

One reason the problems of signed structural node embedding and network classification may be new is because of a lack of benchmark datasets. These formulations are a natural fit for social media analysis, which we perform for our experimental evaluation. We hope our work will also inspire further methodological development as well as introduction of new benchmarks for these problems.

Note: a complete solution for identifying online misbehavior would likely use information beyond the network structure itself, such as text or media content [11]. Our primary goal here is to learn from the network structure *alone*, which has been shown to inform our understanding of social roles [15]. We verify that we capture richer signed network role information than existing graph-based methods.

Social Media Data. We consider two social media datasets: Slashdot Zoo [17] and Cyberbullying [10, 11] on Instagram. We represent each as a network where nodes are users and edges represent pairwise interactions between users. Both datasets contain a subset of users who engage in some sort of online "misbehavior": trolls in Slashdot Zoo, and cyberbullies in Instagram. Intuitively, such socially deviant behavior should manifest itself in a distinctive structural role that we would like to capture in topological feature representations for each user.

In Slashdot Zoo, the edge sign function φ is given *explicitly* by the users themselves, who denote other users as "friends" or "foes" (modeled by positive and negative outgoing edges, respectively). In Cyberbullying, φ must be inferred *implicitly*. The network is defined by users commenting on each other's *media sessions*. Comments are assumed to be directed at the user who posted the picture or video to initiate the session, unless the commenter "mentions" other users using an @ symbol before a username (if so, we form a directed edge from mentioner to mentionee). Thus, φ represents benign or hostile comment intent. Recent preliminary analysis of this dataset [15] found that a strong predictor of a comment's cyberaggression was its score from the VADER model [12] for sentiment analysis in social media, which ranges from -1 (most negative) to 1 (most positive). To avoid misclassification of mildly negative but not truly aggressive comments, we assign an edge sign of 1 for a VADER score above -0.5 and -1 otherwise; we verify this guideline's effectiveness by manual inspection of several comments.

User-Level (Node-Level) Analysis. For the Slashdot dataset, 96 users are marked as trolls by the ground-truth Slashdot account "No More Trolls". We randomly select an equal number of non-trolls and distinguish the two with logistic regression and 10-fold cross validation, trained on various node features:

- **Degree Features**. We concatenate the positive and negative in- and out-degrees of each node to form a four-dimensional feature vector. This is a form of hand engineering using a fundamental structural feature [14] while modeling edge signs.
- **SGCN.** We use the Signed Graph Convolutional Network [2], which performs feature propagation to learn node representations while taking into account balance theory. Such an approach learns community-based node features [20], namely embedding positively-oriented nodes closer than negatively-oriented nodes. This serves as a contrast to our role-based embedding methods.
- Single-sign variations of xNetMF [9]: **xNetMF**$^+$, ignores negative edges and only embeds G^+, while **xNetMF**$^-$ ignores positive edges and only embeds G^-.

We also use signed structural node embeddings via **sNCE** (§ 4.3), and **sec-xNetMF** (§ 4.1): concatenating xNetMF$^+$ and xNetMF$^-$ features. (sRDE's memory requirements are excessive on this larger graph, a limitation shared by its unsigned counterpart GraphWave [13]). All embeddings use the standard dimension $d = 128$ [13].

From the results in Table 1, we see that hand-engineered features (Degree) and features that try to preserve node proximity (SGCN) are the least accurate for the task, which motivates our use of structural node embeddings to characterize troll behavior. Using negative edges alone to determine structural roles leads to slightly better results than using positive edges alone–this makes sense for the task of identifying a negative behavior–but we see that using both positive and negative edges in signed structural embeddings gives the best per-

Table 1: Classifying troll users in Slashdot Zoo. Structural embeddings using both positive and negative edges–in this case disentangling each sign type's effect on the structural role–leads to greatest accuracy.

Method	Accuracy
Degree	0.57
SGCN	0.46
xNetMF$^+$	0.59
xNetMF$^-$	0.61
sec-xNetMF	**0.64**
sNCE	0.51

formance. However, it seems most useful to model the structural roles of users separately in a positive and a negative context, as is evidenced by the worse performance of sNCE and the superiority of sec-xNetMF. Our synthetic experiments (§ 6.1) show that sNCE can detect subtle role differences that sec-xNetMF cannot; however, the differing results on this real dataset may reveal the double-edged nature of this expressivity (e.g. overfitting). Still, we next show that the signed structural embeddings' roles effectively characterize the network itself.

Session-Level (Graph-Level) Analysis. For graph classification, we evaluate the performance of an SVM with 5-fold cross-validation to predict graph's labels using the following kernels or features:

- **Hand-engineered statistics**: We consider the statistical signatures **SMD** and **SDD**, along with the kernel **SGK** discussed in § 5.1.
- **Methods using node embeddings**: we use RGM as discussed in § 5.2 with 16-dimensional signed node embeddings to capture the distribution of structural roles in the network, using sNCE and sRDE respectively.

For this task, we extract the signed who-comments-on-whom networks of 200 Instagram media sessions (§ 3). Each session has one of six ground-truth labels corresponding to the level of cyberbullying it contains [11], which we predict from the network structure. In Table 2, we see that the most powerful predictors are RGM using our structural embeddings. Intuitively, this suggests that the distribution of structural roles as captured by embeddings most informatively characterize the network, more so than statistical signatures or graph kernels designed from hand-engineered features.

Table 2: Classifying the cyberaggression levels occurring in Instagram media sessions. Methods based on signed structural node embedding outperform baselines based on feature engineering.

Method	Accuracy
SMD	0.24
SDD	0.24
SGK	0.23
RGM-sNCE	0.32
RGM-sRDE	**0.33**

7 Conclusion

We have taken a new approach to signed social network mining using node embedding, characterizing nodes based on the structural roles that they play in the signed network. Our methods enable node-level and graph-level analysis that allow us to gain more insights into the social roles of social media users than was previously possible. As the problems we formulated are new, few benchmark datasets or baseline methods exist and we hope that our work will attract more interest to these important problems. Future work may incorporate metadata beyond the network topological structure alone.

Acknowledgments: The authors are grateful to the Defense Advanced Research Projects Agency (DARPA), contract W911NF-17-C-0094, for their support.

References

1. Kai-Yang Chiang, Cho-Jui Hsieh, Nagarajan Natarajan, Inderjit S Dhillon, and Ambuj Tewari. Prediction and clustering in signed networks: a local to global perspective. *JMLR*, 15(1):1177–1213, 2014.
2. Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *ICDM*, 2018.
3. Tyler Derr, Chenxing Wang, Suhang Wang, and Jiliang Tang. Signed node relevance measurements. *arXiv preprint arXiv:1710.07236*, 2017.
4. Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. Learning structural node embeddings via diffusion wavelets. In *KDD*, 2018.
5. Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. In *ICLR*, 2020.
6. Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.
7. Fritz Heider. *The psychology of interpersonal relations*. Psychology Press, 2013.
8. Mark Heimann, Tara Safavi, and Danai Koutra. Distribution of node embeddings as multiresolution features for graphs. In *ICDM*, 2019.
9. Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. Regal: Representation learning-based graph alignment. In *CIKM*, 2018.
10. Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. Analyzing labeled cyberbullying incidents on the instagram social network. In *SocInfo*, 2015.
11. Homa Hosseinmardi, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. Prediction of cyberbullying incidents in a media-based social network. In *ASONAM*, 2016.
12. Clayton J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*, 2014.
13. Di Jin, Mark Heimann, Tara Safavi, Mengdi Wang, Wei Lee, Lindsay Snider, and Danai Koutra. Smart roles: Inferring professional roles in email networks. In *KDD*, 2019.
14. Junchen Jin, Mark Heimann, Di Jin, and Danai Koutra. Understanding and evaluating structural node embeddings. In *KDD MLG Workshop*, 2020.
15. Hsien-Te Kao, Shen Yan, Di Huang, Nathan Bartley, Homa Hosseinmardi, and Emilio Ferrara. Understanding cyberbullying on instagram and ask. fm via social role detection. In *WWW Companion*, 2019.
16. Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
17. Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. The slashdot zoo: mining a social network with negative edges. In *WWW*, 2009.
18. Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *KDD*, 2017.
19. Ryan A Rossi and Nesreen K Ahmed. Role discovery in networks. *TKDE*, 27(4):1112–1131, 2014.
20. Ryan A Rossi, Di Jin, Sungchul Kim, Nesreen K Ahmed, Danai Koutra, and John Boaz Lee. On proximity and structural role-based embeddings in networks: Misconceptions, methods, and applications. *TKDD*, 2020.
21. Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *AISTATS*, 2009.
22. Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. Signed network embedding in social media. In *SDM*, 2017.
23. Shuhan Yuan, Xintao Wu, and Yang Xiang. Sne: signed network embedding. In *PAKDD*, 2017.