CSNE: Conditional Signed Network Embedding

Alexandru Mara
Yoosof Mashayekhi
alexandru.mara@ugent.be
yoosof.mashayekhi@ugent.be
Ghent University
Ghent, Belgium

Jefrey Lijffijt
Tijl de Bie
jefrey.lijffijt@ugent.be
tijl.debie@ugent.be
Ghent University
Ghent, Belgium

ABSTRACT
Signed networks are mathematical structures that encode positive and negative relations between entities such as friend/foe or trust/distrust. Recently, several papers studied the construction of useful low-dimensional representations (embeddings) of these networks for the prediction of missing relations or signs. Existing network embedding methods for sign prediction, however, generally enforce different notions of status or balance theories in their optimization function. These theories, are often inaccurate or incomplete which negatively impacts method performance.

In this context, we introduce conditional signed network embedding (CSNE). Our novel probabilistic approach models structural information about the signs in the network separately from fine-grained detail. Structural information is represented in the form of a prior, while the embedding itself is used for capturing fine-grained information. These components are then integrated in a rigorous manner. CSNE’s accuracy depends on the existence of sufficiently powerful structural priors for modelling signed networks, currently unavailable in the literature. Thus, as a second main contribution, we also introduce a novel approach to construct priors based on the Maximum Entropy (MaxEnt) principle. These priors can model the polarity of nodes (the degree to which their links are positive) as well as signed triangle counts (a measure for the degree structural balance holds to in a network).

Experiments on a variety of real-world networks confirm that CSNE outperforms the state-of-the-art on the task of sign prediction. Moreover, the MaxEnt priors on their own, while less accurate than full CSNE, achieve accuracies competitive with the state-of-the-art at very limited computational cost, thus providing an excellent runtime-accuracy trade-off in resource-constrained situations.

CCS CONCEPTS
• Computing methodologies → Machine learning; Knowledge representation and reasoning.

KEYWORDS
Signed Network Embedding; Sign Prediction; MaxEnt Models
Despite their success, however, unsigned network embedding techniques are not directly applicable to signed networks. Indeed, in signed networks node-pairs can have three possible states: unlinked, positively linked, and negatively linked. Unsigned network embedding methods only consider two possible states (unlinked and linked). Thus, to apply them unaltered to signed networks, one would have to ignore the distinction between two of the three states, e.g. ignoring the difference between negatively linked and unlinked. This is problematic, as the semantics of these states are often very different. For example, unlinked node-pairs are often to be interpreted as having an unknown sign. Moreover, for many signed networks it may be useful to account for balance theory [14] in the model, which is conceptually impossible without distinguishing these three possible states of node-pairs. Balance theory suggests that triads with an odd number of negative connections are unstable and, thus, less likely to form or persist over time. For example, in a trust network structural balance means one is less likely to trust someone who is distrusted by a friend.

To overcome the aforementioned difficulties, several embedding methods for signed networks have been proposed, such as SiNE [38], Signet [15] and SNE [40]. These methods adopt random walk or probabilistic strategies to learn the structure of signed networks and have been shown to outperform unsigned embedding methods for downstream tasks, such as sign prediction. Nevertheless, these methods present two major drawbacks. One hand, they often impose constraints on the learning process that strictly enforce status or balance theories, even though these theories generally only hold to a certain extent [21]. On the other hand, they often suffer from overfitting as only the connected node-pairs are used in the embedding learning process.

In this context, we propose a novel probabilistic approach which we coin Conditional Signed Network Embedding (CSNE). The idea of CSNE is to find an embedding for the nodes in the network that is maximally informative of the signs in the network conditioned on a probabilistic prior for these signs. These two distributions are then combined in a rigorous manner. The rationale is the observation that embeddings are often unsuited for representing complex structural information, while they are strong at representing fine-grained local information [26]. Thus, if we are able to use a probabilistic prior to represent structural information about the network, both types of information can be effectively leveraged for sign prediction. Representing structural and fine-grained information separately has the additional advantage of allowing us to conveniently learn the prior from data, i.e. learn the extent to which structural balance and other properties hold for each network. Moreover, this approach reduces the amount of information the embedding must represent. This, in turn, allows CSNE to use lower embedding dimensionalities, without compromising accuracy, and effectively prevent overfitting.

More specifically, inspired by Conditional Network Embedding (CNE) [17], we construct a likelihood function for the embedding (i.e. a distribution for the signs conditional on the embeddings), and infer the embedding using the Maximum Likelihood principle. The key idea of CSNE is to construct this likelihood function using Bayes rule from a prior distribution for the signs, and a conditional distribution for the embedding conditioned on the signs. This approach allows us to conveniently model structural information in a suitable prior while the embedding itself can focus exclusively on learning fine-grained local connectivity relations between nodes that are more easily expressed in a metric space.

CSNE could work, at least in principle, with any prior for the signs. Yet, in this paper we propose a novel approach for inferring such priors. Our flexible approach can accommodate different types of structural information, however, in this paper we focus on two particular types that have the potential to significantly improve sign prediction performance. The first is that within a network, some nodes may have a more positive (or negative) inclination than others. When modelling, for instance, social interactions, certain users will be liked more often, while others are disliked more often depending on their personality, ideology etc. For new connections formed by these users, we may expect such tendency to persist. We refer to this underlying tendency as the polarity of a node. The second is that structural balance rarely holds exactly, but it may hold to a certain extent. The extent to which it holds can be quantified in terms of the number of balanced triangles (where structural balance holds) versus the number of unbalanced triangles (where it does not hold) in the network. For example, if trust in a trust network is being breached at a high rate, structural balance will hold less strongly. Indeed, even if in an equilibrium situation structural balance would hold, occasional breach of trust can disrupt this balance.

Unfortunately, no approaches have been proposed in the literature modelling such information, for use as a prior in CSNE. Thus, building upon recent work from Adriaens et al. [1] on efficient Maximum Entropy (MaxEnt) models for unsigned graphs, we show how both polarity and the degree to which structural balance holds can be modelled in an accurate and highly scalable manner. This approach is a second main result in this paper.

The main contributions of this paper are:

- Overcoming conceptual and technical challenges we propose a new probabilistic method for signed network embedding and sign prediction. Our method, which we coin Conditional Signed Network Embedding (CSNE) computes the most informative embeddings w.r.t. a prior on the network structure.
- We design a new edge-independent MaxEnt prior specifically for signed networks. This prior models two key structural properties of signed networks: the proportion of positive to negative connections of each node or node polarity and the number of balanced and unbalanced triangles formed by each edge or triangle counts.
- We provide extensive experiments, comparing with existing methods for signed network embedding and sign prediction.

The benefits of these contributions are as follows:

- CSNE is a conceptually novel and mathematically rigorous probabilistic approach to signed network embedding, capable of modelling both structural and fine-grained information. Its modular design means that future research developments on priors for signed networks have the potential to further boost CSNE’s performance.
- The proposed MaxEnt prior, which is conceptually simple and intuitive, is extremely efficient to compute (an order of magnitude faster than SIGNet, the fastest state-of-the-art method for sign prediction). When used on its own for sign
prediction, it already uniformly outperforms all state-of-the-art baselines evaluated across a diversity of commonly used benchmark networks.

- CSNE with the proposed MaxEnt prior further improves upon this accuracy for sign prediction, while requiring computation times equal to or only slightly higher than the fastest state-of-the-art baseline method (SIGNet).

The remainder of the paper is organized as follows. In Sec. 2, we briefly discuss the related work. In Sec. 3, we introduce our proposed method. For ease of exposition, we first discuss the MaxEnt prior, before discussing CSNE and how the prior can be used as a building block in CSNE. Our experimental setup is described in Sec. 4. We report the results in Sec. 5. Finally, Sec. 6 concludes this paper and summarizes open questions.

2 RELATED WORK

Our paper is primarily related to the analysis of signed networks and the prediction of missing signs [18, 34, 38, 40]. Our modular approach, additionally, incorporates ideas from two related fields.

First, CSNE connects to a large body of research in the field of unsigned network representation learning. Early approaches such as Laplacian Eigenmaps [2] and Locally-Linear Embeddings [33] were motivated by dimensionality reduction. More recently, embedding methods have been used to bridge the gap between traditional machine learning and network structured data [3, 24, 30]. Methods such as DeepWalk [32] and Node2vec [11] have been proposed to learn embeddings by leveraging a random walk strategy on the graphs. Other approaches such as GraRep [5] and Arope [41] aim to model high order proximities between nodes in the networks. A recent probabilistic approach, CNE, can efficiently incorporate prior information in the embedding learning process. This method has also been shown in a recent empirical study by Mara et al. [27] to largely outperform other embedding methods for the task of link prediction. For these reasons, our research incorporates ideas introduced in CNE to the context of signed network embedding. As mentioned in the previous section, however, these methods are not directly applicable to the analysis of signed network.

Second, the proposed probabilistic approach for modelling structural properties of signed networks borrows ideas from the field of maximum entropy random graph models [8, 16, 31], and most directly from De Bie [7]. Our approach additionally incorporates recent results from Adriaens et al. [1] to address a fundamental challenge of these models: the difficulty of modelling structural properties, which tend to introduce dependencies between the variables involved, without losing computational tractability.

Several methods for the specific task of signed network embedding have been proposed. One of the first approaches in this field is SNE [40], which computes embeddings through a log-bilinear model that incorporates link types in a given path in the network. More recent approaches [15, 25, 38], impose constraints in the optimization process to enforce different notions of structural balance. SiNE [38] proposes a deep learning framework for signed network embedding and incorporates the extended structural balance theory proposed in Cygan et al. [6]. SIGNet [15] is a scalable node embedding method for signed networks. Akin to CSNE, this method aims to incorporate the notion of balance theory introduced by Heider [14]. Targeted node sampling [29], which extends negative sampling techniques from classical word2vec algorithms, is used to maintain structural balance in higher order neighbourhoods. And in Lu et al. [25], the authors propose SSNE, a method for embedding directed signed networks that considers status theory [12, 22] in its learning phase. The status of a node is determined based on positive and negative links and a ranking is computed. In contrast to these methods, CSNE does not assume a notion of balance must hold exactly. Instead, our method learns the extent to which balance holds for every network. Finally, in another recent work, two methods nSNE and ISNE were proposed for learning node and edge embeddings in signed networks [34].

3 PROPOSED METHOD

In this section we introduce CSNE. We start in Sec. 3.1 by introducing basic concepts and notation for signed networks and sign prediction. In Sec. 3.2, we recap some prior research on MaxEnt graph models and introduce a novel edge-independent MaxEnt distribution that allows one to efficiently model polarity and different types of triangle counts (to model the extent to which structural balance holds) in signed networks. Finally, in Sec. 3.3 we describe the overall CSNE method, and how the MaxEnt model can be used as a building block thereof. In what follows, we limit our analysis exclusively to undirected networks.

3.1 Concepts and notation

3.1.1 Signed Networks. We represent an undirected signed network by \( G = (V,E) \) with \(|V| = n \) nodes and \(|E| = m \) edges where \( E \subseteq \binom{V}{2} \). Each node \( i \in V \) denotes an entity and edges \( \{i,j\} \in E \) represent unordered relations between two entities \( i \) and \( j \). Edges \( \{i,j\} \) can describe positive or negative relations in signed networks. A function \( s : E \rightarrow \{-1, +1\} \) is used to map edges to their respective signs. We denote the set of positive links \( \{(i,j) \in E \mid s(i,j) = 1\} \) by \( E^+ \), and negative links \( \{(i,j) \in E \mid s(i,j) = -1\} \) by \( E^- \). The adjacency matrix of a signed network \( G \) is represented as \( \hat{A} \in \mathcal{A} = \{-1, 0, 1\}^{n \times n} \) with entries \( \hat{a}_{ij} \in \{-1, 0, 1\} \), and with \( \hat{a}_{ij} = 0 \) if \( \{i, j\} \not\in E \) and \( \hat{a}_{ij} = s(i,j) \) otherwise.\(^1\)

3.1.2 Sign Prediction. This task amounts to, given an observed signed network \( G = (V,E) \), inferring the signs \( s \{i,j\} \in \{-1,1\} \) of unobserved node-pairs \( \{i,j\} \in \binom{V}{2} \setminus E \).

3.2 A MaxEnt distribution for edge signs

Maximum entropy random graph models are statistical models used for network inference based on the principle of maximum entropy [16]. This principle states that the best estimate of a distribution given certain constraints is the one with highest entropy amongst those satisfying said constraints. In practice, these constraints are often derived from empirical data.

In the context of unsigned networks, MaxEnt modelling works as follows. Given an observed binary adjacency matrix \( \hat{A} \in \mathcal{A}_{bin} = \binom{V}{2} \)
We already argued that node polarity statistics, and statistics on when fitting MaxEnt models are scalability and generalization to i.e. polarity statistics (one per node), we also use three statistics that corresponding polarity statistic for node \( l \) of each node \( l \) of the signs of edges incident to this node. To model the polarity the number of triangles with particular sign patterns.

To introduce such that their empirical values can be computed. They can also statistics must satisfy the requirement of pertaining to the edges only can be used efficiently for MaxEnt modelling. Note that these sta-

constraints on statistics of the form constraints.

As observed by various authors \([1, 10]\), two major drawbacks in fitting MaxEnt models is to identify statistics that are characteristic for the data, while using them as constraints is computationally tractable. We already argued that node polarity statistics, and statistics on the number of balanced/unbalanced triangles, are both useful to use as constraints. Now, we will show that their use as constraints in MaxEnt modelling is computationally tractable as well. Our approach is based on recent work by Adriaens et al. \([1]\), which identifies a broad class of statistics that can be used to formulate these constraints.

Let \( F \in \mathbb{R}^{n \times n} \) denote a real-valued matrix with \( f_{ij} \) the element on row \( i \) and column \( j \), referred to as a feature matrix. Rephrased for our current context, Adriaens et al. \([1]\) then demonstrate that constraints on statistics of the form

\[
y(A) = \sum_{(i,j) \in E} f_{ij} a_{ij},
\]

(1)
can be used efficiently for MaxEnt modelling. Note that these statistics must satisfy the requirement of pertaining to the edges only such that their empirical values can be computed. They can also be specialised to compute polarity and triangle statistics. Next, we introduce \( n \) feature matrices which we specialize to compute the polarity of each node and three additional matrices for computing the number of triangles with particular sign patterns.

**Polarity.** We formally define the polarity of a node as the sum of the signs of nodes incident to this node. To model the polarity of each node \( l \in V \), we propose to use one feature matrix \( F_l \) as an indicator function of the \( l \)th row, i.e.

\[
f_{ij}^l = \begin{cases} 1, & \text{if } i = l \\ 0, & \text{otherwise}. \end{cases}
\]

(2)

Indeed, for such a feature matrix, \( \sum_{(i,j) \in E} f_{ij}^l a_{ij} = \sum_{j \in E} a_{lj} \), equal to the polarity of node \( l \) as required. Let us denote the corresponding polarity statistic for node \( l \) as \( y_l \).

**Counting triangles with various sign patterns.** In addition to the polarity statistics (one per node), we also use three statistics that jointly describe how many triangles the network contains with three positive signs \((+++\) triangles\), with two positive signs \((++-\) triangles\), with one positive sign \((+-+\) triangles\), and with all negative signs \((-+-\) triangles\). Note that the total number of triangles is fixed, so three of these constraints will suffice. Moreover, using any linear combination of these statistics would be equivalent. In particular, it is mathematically convenient to count the following three statistics which jointly can be used:

- **Triangle statistic** \( y_{++} \) This statistic is defined as the number of +++ triangles minus the number of +++ triangles. This can be counted in the form of Eq. (1) by defining \( f_{ij} \) as the number of wedges, or paths of length two between \( i \) and \( j \), with two positive signs, or formally:

\[
f_{ij}^{++} = \sum_{k:(k,i) \in E \land (k,j) \in E} \hat{a}_{ik} + \hat{a}_{kj} + \frac{1}{2}.
\]

(3)

Then \( \sum_{(i,j) \in E} f_{ij}^{++} \hat{a}_{ij} = \sum_{(i,j) \in E} f_{ij}^{++} - \sum_{(i,j) \in E} f_{ij}^{-+} + \frac{1}{2} \), which is indeed equal to the number of +++ triangles minus the number of +++ triangles.

- **Triangle statistic** \( y_{+} \) This statistic is defined as the number of +++ triangles minus the number of ++- triangles. It can be computed as in Eq. (1) by defining \( f_{ij}^{+-} \) as the number of wedges with differing signs that connect \( i \) and \( j \), or formally:

\[
f_{ij}^{+-} = \sum_{k:(k,i) \in E \land (k,j) \in E} 1 - \frac{1}{2} \hat{a}_{ik} \hat{a}_{kj}. \]

(4)

- **Triangle statistic** \( y_{-} \) This statistic is defined as the number of +++ triangles minus the number of -- triangles. It can be computed as in Eq. (1) by defining \( f_{ij}^{-} \) as the number of wedges with two minus signs connecting \( i \) and \( j \), or formally:

\[
f_{ij}^{-} = \sum_{k:(k,i) \in E \land (k,j) \in E} 1 - \frac{1}{2} \hat{a}_{ik} \hat{a}_{kj}. \]

(5)

Note that these three statistics jointly, together with the fixed overall triangle count, indeed uniquely define the number of triangles with any sign pattern. For example, denoting the total number of triangles as \( t \), it is easy to verify that the number of +++ triangles can be computed as \( t_{++} = \frac{1}{2} t - 2 t_{+-} + t_{-} \).

Thus the MaxEnt distribution can be found by solving the following convex optimization problem:

\[
\arg\max_{P(A)} -\mathbb{E}_P[\log P(A)],
\]

(6)

s.t. \( \mathbb{E}_P[y_l(A)] = c_l \) \( \forall l = 1, \ldots, n \),

\( \mathbb{E}_P[y_{++}(A)] = c_{++} \),

\( \mathbb{E}_P[y_{+}(A)] = c_{+} \),

\( \mathbb{E}_P[y_{-}(A)] = c_{-} \),

where \( c_l = y_l(\hat{A}) \), \( c_{++} = y_{++}(\hat{A}) \), \( c_{+} = y_{+}(\hat{A}) \), and \( c_{-} = y_{-}(\hat{A}) \) denote the empirically observed values of the statistics.

As shown by Adriaens et al. \([1]\), the formulation in Eq. (6) factorizes as a product of independent Bernoulli distributions. The solution, adapted to the case of signed networks, is of the form:

\[
P(A) = \prod_{(i,j) \in E} \frac{\hat{a}_{ij}^{y_{ij}^{+-}} (1 - P(aij = 1))^{1 - \hat{a}_{ij}^{y_{ij}^{+-}}}}{\hat{a}_{ij}^{y_{ij}^{++}} (1 - P(aij = 1))^{1 - \hat{a}_{ij}^{y_{ij}^{++}}}}.
\]

(7)
where $\lambda_1, \lambda_{++}, \lambda_-$, and $\lambda_{--}$ denote the Lagrange multipliers associated with the respective constraints in Eq. (6). The optimal values of these Lagrange multipliers can be found by unconstrained minimization of the convex Lagrange dual function:

$$L(\lambda_1, \ldots, \lambda_M) = \sum_{\{i,j\} \in E} \log(1 + \exp(\sum_{l=1}^{M} f_{ij}^l \lambda_l + f_{ji}^{++} \lambda_{++} + f_{ji}^{--} \lambda_{--} + f_{ij}^{--} \lambda_{--})) - \sum_{l=1}^{M} c_l \lambda_l - c_{++} \lambda_{++} - c_{--} \lambda_{--} - c_{-} \lambda_{-}.$$  

This optimization problem can be solved efficiently for very large networks, as shown by Adriaens et al. [1], by using equivalences between the Lagrange multipliers. In our proposed model, this reduces the number of free variables to $O(\sqrt{N})$. Moreover, we can further improve efficiency by limiting our computations to the connected node-pairs for which signs are known. Finally, we observe that after fitting the MaxEnt prior, a probability for each node-pair of being positive or negative is obtained. These unique probabilities, determined by the polarities of the connected nodes and triangle counts, can be directly used for sign prediction.

### 3.3 Conditional Signed Network Embedding

As introduced in Sec. 1, the MaxEnt model is required as a prior for CSNE. For ease of exposition, the prior was discussed in the previous section. In the current section, we introduce CSNE proper and explain how the MaxEnt prior is an integral component. However, before diving into the details of CSNE, we formally define the task of signed network embedding.

Embedding approaches learn a function $g : V \rightarrow \mathbb{R}^d$ which maps nodes in the network to $d$-dimensional real-valued vectors. These representations are generally denoted as $X = (x_1, x_2, \ldots, x_N)^T \in \mathbb{R}^{N \times d}$, where $x_i$ is the embedding corresponding to node $i$. A common modelling assumption for NE methods is that similar nodes in the network must be mapped to close-by representations in the embedding space. For the particular case of signed network embedding, we additionally require the function $g : V \rightarrow \mathbb{R}^d$ to map pairs $\{i, j\} \in E^+$ to more distant representations $x_i$ and $x_j$ in the embedding space, than pairs $\{i, j\} \in E^-$. At the same time, unlinked node-pairs are commonly not used in the embedding learning process i.e. their distance is unimportant.

CSNE aims to learn the most informative embedding $X$ for a given signed network $G = (V, E)$ with adjacency matrix $\hat{A}$. In simple terms, the objective is to find the embedding $X$ that maximizes the likelihood of observing the signs on the edges in $\hat{A}$. We formulate this optimization task as a Maximum Likelihood Estimation (MLE) problem, $\operatorname{argmax}_X P(A|X)$. Akin to Kang et al. [17], we do not postulate the likelihood function $P(A|X)$ directly. Instead, we do postulate the density function of the embedding $X$ conditioned on the signed network, i.e. $p(X|A)$. Then, we combine $p(X|A)$ with the MaxEnt prior discussed in Sec. 3.2, $P(A)$, by means of the Bayes formula. The likelihood function, thus, follows as $P(A|X) = \frac{p(X|A)p(A)}{p(X)}$. Independently modelling the density function of the embedding conditioned on the network and the MaxEnt distribution on the signs of $G$, has one major advantage. The prior, can encode properties of the network which do not have to be reflected by the learned embedding $X$. Effectively, this means that CSNE can make a better use of the embedding space.

The MaxEnt prior $P(A)$ has already been introduced in Sec. 3.2 and its expression given in Eq. 7. For postulating the conditional density $p(X|A)$, we only need to model the distances between connected node-pairs $\{i, j\}$. As metric, we use the Euclidean distance between the embeddings of the end-nodes, i.e. $d_{ij} = ||x_i - x_j||_2$. As already discussed, we require the distances between positively connected node-pairs $\{i, j\} \in E^+$ to be lower than those between negatively connected pairs $\{i, j\} \in E^-$. To model this, we use two half-normal distributions with locations at 0 and different spread parameters. Distances between pairs $\{i, j\} \in E^+$ are generated from a half-normal distribution with spread parameter $\sigma_1$ and between pairs $\{i, j\} \in E^-$ from a similar distribution with parameter $\sigma_2$, where $\sigma_1 < \sigma_2$. Thus, we have:

$$p(d_{ij} | \{i, j\} \in E^+) = \mathcal{N}_+(d_{ij}|\sigma_1^2), \quad (10)$$

$$p(d_{ij} | \{i, j\} \in E^-) = \mathcal{N}_+(d_{ij}|\sigma_2^2). \quad (11)$$

The conditional density $p(X|A)$ can then be expressed as follows:

$$p(X|A) = \prod_{\{i,j\} \in E^+} \mathcal{N}_+(d_{ij}|\sigma_1^2) \cdot \prod_{\{i,j\} \in E^-} \mathcal{N}_+(d_{ij}|\sigma_2^2). \quad (12)$$

The resulting likelihood function to optimize is:

$$P(A|X) = \frac{p(X|A)p(A)}{p(X)} = \frac{p(X|A)p(A)}{\sum_A p(X|A)p(A)} \mathcal{N}_+(d_{ij}|\sigma_1^2)^{P(a_{ij} = 1)} \cdot \prod_{\{i,j\} \in E^-} \mathcal{N}_+(d_{ij}|\sigma_2^2)^{P(a_{ij} = 1)} + \mathcal{N}_+(d_{ij}|\sigma_1^2)^{P(a_{ij} = 1)} \cdot \prod_{\{i,j\} \in E^-} \mathcal{N}_+(d_{ij}|\sigma_2^2)^{P(a_{ij} = 1)} \cdot \prod_{\{i,j\} \in E^-} \mathcal{N}_+(d_{ij}|\sigma_1^2)^{P(a_{ij} = 1)} \cdot \prod_{\{i,j\} \in E^-} \mathcal{N}_+(d_{ij}|\sigma_2^2)^{P(a_{ij} = 1)}. \quad (13)$$

In order to maximize the likelihood function Eq. (13) we use block stochastic gradient descent. The derivation of the gradient follows closely the one in Kang et al. [17]. Thus, we refer the user to this manuscript for more details. The gradient of the log-likelihood function in CSNE w.r.t. the embedding $x_i$ of node $i$ reads:

$$\frac{\partial \log(P(A|X))}{\partial x_i} = 2 \sum_{j: \{i,j\} \in E^+} (x_i - x_j) P(a_{ij} = 1 | X) \left( \frac{1}{\sigma_1^2} - \frac{1}{\sigma_1^2} \right) + 2 \sum_{j: \{i,j\} \in E^-} (x_i - x_j) P(a_{ij} = 1 | X) \left( \frac{1}{\sigma_2^2} - \frac{1}{\sigma_2^2} \right). \quad (14)$$

Intuitively, the first summation in the gradient expression pulls the embedding of each node close to the embeddings of positively
We performed sign prediction evaluation using 6 different real-world datasets. The first two networks constitute snapshots from the Slashdot [22] social news website, where users can create friends (positive links) and foes (negative links). These snapshots, Slashdot(a) and Slashdot(b), were obtained in November 2008 and February 2009, respectively. Epinions [22] is a product review website, where users can trust (positive links) or distrust (negative links) each other. Wiki-rfa [39] contains votes of Wikipedia users endorsing or opposing candidates for adminship. Neutral votes, also present in the data, are not used. The dataset contains information on votes between 2003 and 2013. Lastly, we used two Bitcoin cryptocurrency trust networks, Bitcoin-ar and Bitcoin-otec. These networks, obtained from [19], were gathered in order to identify transactions with fraudulent users.

All datasets, which were originally directed, were preprocessed to remove the direction of links. In the experiments, we also used the largest connected components only and ignored self-loops. The most relevant network statistics are summarized in Table 1.

<table>
<thead>
<tr>
<th>Data</th>
<th>Slashdot(a)</th>
<th>Slashdot(b)</th>
<th>Epinions</th>
<th>Wiki-rfa</th>
<th>Bitcoin-ar</th>
<th>Bitcoin-otec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>77350</td>
<td>82140</td>
<td>131828</td>
<td>11258</td>
<td>3783</td>
<td>5881</td>
</tr>
<tr>
<td></td>
<td>468554</td>
<td>500481</td>
<td>712120</td>
<td>171562</td>
<td>14124</td>
<td>21492</td>
</tr>
<tr>
<td></td>
<td>12.11</td>
<td>12.18</td>
<td>10.78</td>
<td>30.47</td>
<td>7.46</td>
<td>7.30</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>76%</td>
<td>83%</td>
<td>77%</td>
<td>90%</td>
<td>85%</td>
</tr>
<tr>
<td></td>
<td>85%</td>
<td>86%</td>
<td>89%</td>
<td>73%</td>
<td>83%</td>
<td>85%</td>
</tr>
</tbody>
</table>

Table 1: Main statistics of the networks used for sign prediction evaluation.

4 EXPERIMENTAL SETUP

To evaluate the performance of CSNE, we conducted experiments on six networks from four sources, which we introduce in Sec. 4.1. We compared the empirical observations for CSNE with four recent methods for signed network embedding that are discussed in Sec. 4.2. Specifics of the test setup, including hyperparameters settings for all methods are outlined in Sec. 4.3. Reproducibility details are provided in Sec. 4.4.

4.1 Datasets

We performed sign prediction evaluation using 6 different real-world datasets. The first two networks constitute snapshots from the Slashdot [22] social news website, where users can create friends (positive links) and foes (negative links). These snapshots, Slashdot(a) and Slashdot(b), were obtained in November 2008 and February 2009, respectively. Epinions [22] is a product review website, where users can trust (positive links) or distrust (negative links) each other. Wiki-rfa [39] contains votes of Wikipedia users endorsing or opposing candidates for adminship. Neutral votes, also present in the data, are not used. The dataset contains information on votes between 2003 and 2013. Lastly, we used two Bitcoin cryptocurrency trust networks, Bitcoin-ar and Bitcoin-otec. These networks, obtained from [19], were gathered in order to identify transactions with fraudulent users.

All datasets, which were originally directed, were preprocessed to remove the direction of links. In the experiments, we also used the largest connected components only and ignored self-loops. The most relevant network statistics are summarized in Table 1.

4.2 Comparison methods

To be able to interpret the performance of CSNE in the context of existing methods, we performed the same sign prediction experiments using the following methods:

- SiNE [38], similarly to SiNE, uses a deep neural network to learn node embeddings by leveraging second order proximities in the graph. At the same time, the method learns a mapping from node embeddings to edge embeddings.
- ISNE [34] is a simplified version of nSNE where the mapping function from node embeddings to edge embeddings is assumed to be linear. The function can therefore be learned via gradient descent.
- SIGNet [15] uses a random walk strategy to determine node similarity on the graph and the Skip-Gram model to obtain node embeddings. The authors propose an extension to the negative sampling used in word2vec models to perform targeted node sampling and accommodate the main concepts of structural balance theory (i.e., balanced triangles are more likely than unbalanced triangles).

4.3 Sign Prediction Test Setup

General setup. As introduced in Sec. 3.1.2, sign prediction amounts to identifying the signs of unobserved connections between nodes in a given network. For performance evaluation of sign prediction, it is common to divide the given set of edges (E) into two disjoint subsets: the train edges (E_train) are used in the model learning phase, while the test edges (E_test) are used for assessing the prediction performance of the methods. The train and test sets are constructed such that E_test \cup E_train = E, and E_test \cap E_train = \emptyset. At training time, edges in E_train are removed and the corresponding value in the adjacency matrix is set to 0. At test time, the model is evaluated on all \{i, j\} \in E_test and predictions compared to \hat{a}_{ij}. In our evaluation, we selected 80% of the total edges \{i, j\} \in E for training (E_train) and the remaining 20% for testing (E_test). Train edges were selected regardless of their sign, using the default sampling strategy of the EvaNE toolbox Mara et al. [28], which ensures that the training network remains connected. The sets E_train and E_test are expected to contain similar proportions of positive and negative edges as the original graph. Unless otherwise specified, all results reported are averages over three independent repetitions of the experiment with different train and test sets.

Edge embeddings. For prediction, nSNE, ISNE and our proposed methods can directly return the probability of an edge \{i, j\} of being positive or negative. For SiNE and SIGNet, however, this is not the case. These methods only return node embeddings from which predictions must be derived. As shown by Gurukar et al. [13] an effective approach for obtaining predictions, in this case, is through binary classification on the edge embeddings derived from node embeddings. A classifier is first trained on the embeddings of edges in E_train and their corresponding signs \{-1, 1\}. Then, predictions for node-pairs in E_test can be computed. To obtain the embedding of a link \{i, j\} from node representations one can apply different operators \circ to the embeddings of the incident nodes i and j i.e. x_{ij} = x_i \circ x_j. Our evaluation, we selected the operators introduced in Grover and Leskovec [11], namely Average ((x_i + x_j)/2), Hadamard (x_i \cdot x_j), Weighted L1 \|x_i - x_j\|_1, and Weighted L2 \|x_i - x_j\|^2. The choice of operator was tuned as additional method hyperparameters for SiNE and SIGNet and Logistic Regression with 5 fold cross validation of the regularization parameter was used as binary classifier.
We performed method hyperparameter tuning on a validation set well suited for prediction in the case of class imbalance. Let $TP$, $FN$, $TN$ and $FP$ denote the number of true positives, false negatives, true negatives and false positives respectively. To guarantee the reproducibility of our experimental evaluation we learned by CSNE on a small network representing relations between characters in the Harry Potter novels.

5 EXPERIMENTAL RESULTS

In this section we present and discuss quantitative and qualitative experimental results. Quantitative results, on one hand, are shown for the task of sign prediction. A qualitative evaluation, on the other, is performed through visualization of the signed embeddings learned by CSNE on a small network representing relations between characters in the Harry Potter novels.

### 5.1 Sign Prediction

We start in Table 2 by presenting the AUC scores for each method on all evaluated datasets. For CSNE we present the results using two MaxEnt priors with different sets of constraints, i.e. node polarity only (pol.) and node polarity combined with balanced/unbalanced triangle counts (pol.+tri.). These MaxEnt priors, as discussed in Sec. 3.2, can be independently used for sign prediction. Therefore, we also include their performance in Table 2 as Prior(pol.) and Prior(pol.+tri.), respectively.

We start in Table 2 by presenting the AUC scores for each method on all evaluated datasets. For CSNE we present the results using two MaxEnt priors with different sets of constraints, i.e. node polarity only (pol.) and node polarity combined with balanced/unbalanced triangle counts (pol.+tri.). These MaxEnt priors, as discussed in Sec. 3.2, can be independently used for sign prediction. Therefore, we also include their performance in Table 2 as Prior(pol.) and Prior(pol.+tri.), respectively.

The results in Table 2 also show that using balanced and unbalanced triangle counts as additional structural constraints to the node polarity in the MaxEnt prior, always results in improved AUC scores. This effect is most prominent when comparing Prior(pol.) and Prior(pol.+tri.), where the latter obtains higher accuracy scores across the board. A similar, yet less prominent effect can be observed when these priors are used as part of CSNE. Another interesting observation from Table 2 is that the proposed Prior(pol.+tri.) already provides state-of-the-art results while being much faster than other methods, as shown by our runtime experiments.

In Figure 1 we present the execution times in seconds (all experiments were run on a machine equipped with an Intel(R) Core i7-7700K processor and 32GB of RAM), including hyperparameter tuning, for all methods. For CSNE, the results include prior computation. Each colour in the figure represents a different network and the grey boxes indicate, per method, the corresponding cumulative execution times on all networks (i.e. sum of all coloured bars). An immediate observation from Figure 1 is that the total execution time of SiNE (grey bar in the Figure 1), is approximately two orders of magnitude larger than those of other methods. The cumulative execution times on all networks of the two MaxEnt priors, on the other hand, are approximately one order of magnitude lower than those of the fastest evaluated baseline, SIGNet. We also observe

<table>
<thead>
<tr>
<th>Methods</th>
<th>Slashdot(a) AUC</th>
<th>Slashdot(b) AUC</th>
<th>Epinions AUC</th>
<th>Wiki-rfa AUC</th>
<th>Bitcoin-α AUC</th>
<th>Bitcoin-otc AUC</th>
<th>Avg. AUC Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiNE</td>
<td>0.850±0.002</td>
<td>0.856±0.002</td>
<td>0.898±0.001</td>
<td>0.816±0.002</td>
<td>0.835±0.012</td>
<td>0.857±0.005</td>
<td>8</td>
</tr>
<tr>
<td>nSiNE</td>
<td>0.895±0.001</td>
<td>0.894±0.002</td>
<td>0.950±0.001</td>
<td>0.879±0.002</td>
<td>0.810±0.024</td>
<td>0.868±0.010</td>
<td>4.5</td>
</tr>
<tr>
<td>ISNE</td>
<td>0.886±0.002</td>
<td>0.893±0.002</td>
<td>0.941±0.002</td>
<td>0.872±0.001</td>
<td>0.854±0.022</td>
<td>0.907±0.005</td>
<td>6</td>
</tr>
<tr>
<td>SIGNet</td>
<td>0.887±0.002</td>
<td>0.893±0.002</td>
<td>0.943±0.001</td>
<td>0.848±0.005</td>
<td>0.893±0.002</td>
<td>0.928±0.006</td>
<td>4.5</td>
</tr>
<tr>
<td>Prior(pol.)</td>
<td>0.879±0.001</td>
<td>0.885±0.001</td>
<td>0.929±0.001</td>
<td>0.870±0.001</td>
<td>0.858±0.014</td>
<td>0.891±0.002</td>
<td>7</td>
</tr>
<tr>
<td>Prior(pol.+tri.)</td>
<td>0.888±0.001</td>
<td>0.894±0.001</td>
<td>0.937±0.001</td>
<td>0.875±0.001</td>
<td>0.874±0.009</td>
<td>0.914±0.004</td>
<td>3</td>
</tr>
<tr>
<td>CSNE(pol.)</td>
<td>0.902±0.001</td>
<td>0.908±0.001</td>
<td>0.952±0.001</td>
<td>0.886±0.002</td>
<td>0.896±0.009</td>
<td>0.930±0.002</td>
<td>2</td>
</tr>
<tr>
<td>CSNE(pol.+tri.)</td>
<td>0.904±0.001</td>
<td>0.909±0.001</td>
<td>0.954±0.000</td>
<td>0.887±0.002</td>
<td>0.899±0.006</td>
<td>0.936±0.002</td>
<td>1</td>
</tr>
</tbody>
</table>
that the two CSNE variants are not significantly slower, on most networks, than other baseline methods.

In Figure 2 we group the methods by dataset and present their execution times relative to the fastest method in each case. Firstly, we observe that Prior(pol.) is the fastest method on all networks and that Prior(pol.+tri.) is never more than 2x slower. The remaining methods, with the exception of SiNE, are between 10x and 50x slower. For SiNE, we observe execution times that are up to 2000 times those of our proposed Prior(pol.) approach on the two Slashdot and the Epinion networks. Finally, we observe that the relative speedup of Prior(pol.) over the baselines becomes smaller as the network sizes increase. This effect is due to parallelization, present in the baseline implementations and not in the MaxEnt prior, which becomes more relevant for larger networks. Parallelization of the MaxEnt prior is left as future work.

5.2 Hyperparameter Sensitivity

We evaluated the sensitivity of the proposed CSNE method w.r.t. three hyperparameters, i.e. train set size, embedding dimensionality and the spread parameter $\sigma$.

5.2.1 Train Set Size. First, we assessed the generalization performance of the proposed method from different amounts of initial training data. We did this by dividing the set of all graph edges $E$ in sets $E_{train}$ and $E_{test}$ of different sizes. We started by using 35% of all edges for training and 65% for testing. We then gradually increased the size of $E_{train}$ to 50%, 65% and finally 80% while the size of $E_{test}$ scaled accordingly 50%, 35% and 20%. Using the same setting as in Sec. 4.3 we performed sign prediction evaluation for these different edge splits. For comparison, we also included the AUC scores of the baseline methods. The average AUC scores of each method over all evaluated networks, are summarized in Figure 3. The 95% confidence intervals are also presented for each case.

The results show that the performance of lSNE, nSNE and SIGNet degrades significantly as the size of $E_{train}$ decreases. CSNE, the MaxEnt prior and SiNE, on the other hand, are more robust to changes in the size of $E_{train}$. The tight confidence intervals for these three methods also indicate a consistent performance across different datasets. For lSNE, nSNE and SIGNet these ranges are larger, especially when little training data is available.

5.2.2 Embedding Dimensionality. Another fundamental parameter for signed network embedding methods is the size of the resulting embeddings. We studied the performance of CSNE w.r.t. the dimensionality for $d \in \{2, 4, 8, 16, 32\}$. The results, depicted in Figure 4a, show a consistent performance of the method for all values of this parameter. Even for values as low $d = 2$, the performance of our method is excellent. This indicates that CSNE can be directly used for visualization without the need to use additional tools.

5.2.3 Spread parameter $\sigma$. The two $\sigma$ parameters introduced in Sec. 3.3 determine, on one hand, the spread of the Gaussian distributions from which the distances between positively and negatively connected pairs are generated in Eq. (12). These parameters also control the strength of the pull and push effects in Eq. (14). Therefore, to understand the effect on performance of different values for these parameters, we conducted and additional experiment. In this experiment, we set $\sigma_1 = 1$, as this simply fixes the scale, and varied $\sigma_2 \in \{2, 4, 8, 16, 32\}$. The results, summarized in Figure 4b, show the robustness of CSNE to changes of this parameter. Only
Figure 4: Sign prediction AUC scores of CSNE(pol.+tri.) for (a) different embedding dimensions $d$ and (b) different values of $\sigma_2$.

Figure 5: Convergence of (a) the MaxEnt prior and (b) CSNE. Very large values of $\sigma_2$, i.e., $\sigma_2 = 16$ and $\sigma_2 = 32$, appear to have a slight effect on method performance. As such, $\sigma_1 = 1$ and $\sigma_2 = 2$ appear to be good default values in most cases.

5.3 Convergence Analysis

In this section we present and discuss the convergence of the MaxEnt prior and of the complete CSNE approach. In Figures 5a and 5b we plot, for each method, the sign prediction AUC against the gradient descent iteration number on all datasets introduced in Table 1. In both cases, we used the joint polarity and triangle count prior. Computing the MaxEnt prior amounts to optimizing a convex function as discussed in Sec. 3.2. By leveraging second order information, convergence is achieved in less than 20 iterations for all datasets as shown in Figure 5a. In CSNE, maximizing the likelihood function is a non-convex optimization problem solved via block stochastic gradient descent. Convergence is achieved in approximately 200 iterations as shown in Figure 5b.

5.4 A Case Study: Visualization

We also performed a qualitative evaluation of the CSNE embeddings on the popular Harry Potter network. Nodes in the graph correspond to characters in the novels while edges denote friend or enemy relations extrapolated from the character interactions throughout the novels. We preprocessed the original directed network to obtain an undirected representation, extracted the main connected component and removed self loops. The resulting network $G$ contained $n = 65$ nodes and $m = 453$ edges with an average degree of 14.

Figure 6: Plot showing the CSNE embeddings of the Harry Potter network. Enemy relationships between book characters are highlighted in red, while friendship is highlighted in blue. A subset of protagonists and antagonists are shown in blue and red, respectively.

We computed a 2-dimensional embedding $X$ of $G$ using CSNE with a MaxEnt structural prior encoding node polarity and the number of balanced and unbalanced triangles. We used 100 iterations to fit the prior and another 100 iterations to learn the embeddings. We obtained an initial assessment of the embedding quality by performing sign prediction for all edges $E$ of $G$. The resulting AUC score is 0.994.

The embeddings learned by CSNE are presented in Figure 6 where blue links denote friendship relations and red links denote enemy relations. The main protagonists and antagonists of the novels are presented as blue and red circles, respectively. Two clear clusters can be identified, a larger one corresponding to the protagonists and allies and a smaller one for the antagonists. The relations within each cluster are mostly positive while between clusters are negative. To verify that, as expected, positively connected nodes are, on average, closer to each other than negatively connected ones, we used the Euclidean distance. The average Euclidean distance obtained for all positively connected pairs $\{i, j\} \in E^+$ is 0.745 with a standard deviation of 0.525. For the negatively connected pairs, $\{i, j\} \in E^-$, the obtained distance is $3.360 \pm 1.014$. This experiment shows that CSNE is able to effectively capture the structure of a signed network in dimensionalities as low as $d = 2$. This also showcases the potential of CSNE for signed network visualization.

6 CONCLUSION AND FUTURE WORK

In this paper, we have presented a new probabilistic approach for learning representations of signed networks with applications to visualization and sign prediction. The proposed CSNE method solves an MLE problem which seeks the embeddings that maximize the probability of observing the signs on the edges of an input graph. Our optimization process models certain structural properties of the data as a MaxEnt prior. Particularly, this prior captures node polarity and structural balance i.e. as counts of balanced and unbalanced triangles. Our experimental results indicate that CSNE can adequately model the specific properties of signed networks and outperforms other baselines for sign prediction. Additionally,
we have showed that the proposed MaxEnt priors can also be directly used for sign prediction, resulting in state-of-the-art AUC scores with runtimes up to 50x lower than those of other baselines.

Our work opens up several avenues for further research and improvements. On one hand, more sophisticated MaxEnt priors can be designed specifically for networks with particular structures, such as k-partition. Another possible line of work is to tackle the extension of CSNE to directed signed networks.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme, and from the FWO (project no. G091017N, G09816N, G042220).

REFERENCES