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, which we find to be highly valuable in its own right, 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

1 INTRODUCTION

In recent years, signed networks have become prominent in online media representing friend/foe relations [20], trust networks describing trust/distrust [12], natural language processing for modelling synonyms and antonyms [15], and sentiment analysis denoting positive or negative opinions [36]. Signed networks are powerful data representations describing complex interactions between heterogeneous entities represented as nodes (or vertices). Generally, we refer to these interactions or node-pair connection as edges or links. In contrast to classical networks, where links simply denote a relation between entities, in signed networks these relations can be 'positive' (e.g. friend, trust) or 'negative' (e.g. foe, distrust).

An important task on signed networks is the prediction of signs between node-pairs for which this sign is unknown. This contrasts with link prediction, where the purpose is to predict whether any given node-pair with unknown link status should be linked or not. Sign prediction is an important problem in practice owing to the high cost of acquiring information on the signs in many domains. In social networks, for example, interactions between users (e.g. messages exchanged, or connections made) can be efficiently tracked. Unveiling the positive or negative nature of the relations between users (i.e. edge signs) from these interactions, however, is more challenging. In a trust network it may be easy to keep track of who engaged in a transaction with whom, but much less easy to understand the mutual trust relations underlying these transactions. Note that sign prediction not only allows one to predict the sign of a link in the network, it also allows one to predict the sign of a link the existence of which is not established. In social networks, for example, this would allow one to assess whether two people are likely to become friends or foes if introduced to each other. Sign prediction has thus been the topic of many recent studies [18, 34, 38, 40].

A particular class of approaches extensively used for the analysis of *unsigned* networks consists of representation learning methods, also called network embedding methods [17, 32]. These techniques model nodes as low-dimensional vectors in $\mathbb{R}^d$. The underlying idea is that similar nodes in the graph are mapped to close-by vectors in the embedding space. Using these representations, traditional machine learning methods can be applied on network data to perform downstream tasks such as link prediction [11, 37], information diffusion [4, 9, 23], and multi-label classification [35].

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: un-linked, 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 distin-guishing 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. On 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 ap-proach 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 litera-ture 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 in-formative 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, capa-ble of modelling both structural and fine-grained informa-tion. Its *modular design* means that future research develop-ments 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 lSNE 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.e. $\{\{i, j\} \in E \mid s(\{i, j\}) = 1\}$ 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 $\{i, j\} \notin E$ 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 $\hat{A} \in \mathcal{A}_{bin} =$

---

[1]In some settings, it might be useful to consider a fourth possible state of a node-pair $\{i, j\} \in \binom{V}{2}$: linked (i.e. $\{i, j\} \in E$), but with unknown sign (i.e. $s(\{i, j\})$ unknown). Although semantically this is different, from a methodological perspective the distinction between such node-pairs and unlinked node-pairs is irrelevant, as the only node-pairs used for training in any existing method (including ours) are those for which the sign is known.

$\{0, 1\}^{n \times n}$ and a set of statistics (e.g. node degrees, assortativity, densities of particular blocks, etc.) considered characteristic for this network, one seeks the MaxEnt distribution P over random $\mathbf{A} \in \mathcal{A}_{bin}$ such that the expected values of these statistics are equal to the values empirically observed on $\hat{\mathbf{A}}$. For instance, one can seek the MaxEnt distribution P which preserves the node degrees of $\hat{\mathbf{A}}$ i.e. $\mathbb{E}_P[\sum_i \mathbf{A}_{i,j}] = \sum_i \hat{\mathbf{A}}_{i,j}$.

Here, our objective is to learn a MaxEnt distribution $P(\mathbf{A})$ for the *signs* of all node-pairs $\{i, j\} \in \binom{V}{2}$. We use a similar formulation to that of MaxEnt models for unsigned graphs, with a key difference: the random variables are now the signs $s(\{i, j\})$ instead of whether a node-pair is connected or not. In summary, we aim to find a distribution over the set $\mathcal{A}_{sign} = \{-1, 1\}^{n \times n}$ of all possible signed matrices $\mathbf{A}$ of size n, with expected values for certain important statistics equal to their values on the empirically observed signs. Of course, as the signs are known only for the linked node-pairs, the statistics are computed based on the linked node-pairs only.

As observed by various authors [1, 10], two major drawbacks when fitting MaxEnt models are scalability and generalization to arbitrary constraints. Thus, a key challenge in the development of 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 $\mathbf{F} \in \mathbb{R}^{n \times n}$ 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

$$\gamma(\mathbf{A}) \triangleq \sum_{\{i,j\} \in E} f_{ij} a_{ij}, \qquad (1)$$

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 patters.

Polarity. We formally define the polarity of a node as the sum of the signs of edges incident to this node. To model the polarity of each node $l \in V$, we propose to use one feature matrix $\mathbf{F}_l$ as an indicator function of the lth row, i.e.

$$f_{ij}^l = \begin{cases} 1, & \text{iff } i = l \\ 0, & \text{otherwise.} \end{cases} \qquad (2)$$

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

$$f_{ij}^{++} = \sum_{k:\{k,i\} \in E \wedge \{k,j\} \in E} \frac{\hat{a}_{ik} + 1}{2} \frac{\hat{a}_{kj} + 1}{2}. \qquad (3)$$

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

- **Triangle statistic γ_{+-}** 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:

$$f_{ij}^{+-} = \sum_{k:\{k,i\} \in E \wedge \{k,j\} \in E} \frac{1 - \hat{a}_{ik} \hat{a}_{kj}}{2}. \qquad (4)$$

- **Triangle statistic γ_{--}** 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:

$$f_{ij}^{--} = \sum_{k:\{k,i\} \in E \wedge \{k,j\} \in E} \frac{1 - \hat{a}_{ik}}{2} \frac{1 - \hat{a}_{kj}}{2}. \qquad (5)$$

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 $\frac{t + \gamma_{--} + 2\gamma_{+-} - \gamma_{++}}{4}$.

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

$$\underset{P(\mathbf{A})}{\arg\max} \quad -\mathbb{E}_P[\log P(\mathbf{A})], \qquad (6)$$

$$\text{s.t.} \quad \mathbb{E}_P[\gamma_l(\mathbf{A})] = c_l \quad \forall l = 1, \dots, n,$$
$$\mathbb{E}_P[\gamma_{++}(\mathbf{A})] = c_{++},$$
$$\mathbb{E}_P[\gamma_{+-}(\mathbf{A})] = c_{+-},$$
$$\mathbb{E}_P[\gamma_{--}(\mathbf{A})] = c_{--},$$

where $c_l = \gamma_l(\hat{\mathbf{A}})$, $c_{++} = \gamma_{++}(\hat{\mathbf{A}})$, $c_{+-} = \gamma_{+-}(\hat{\mathbf{A}})$, and $c_{--} = \gamma_{--}(\hat{\mathbf{A}})$ 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:

$$P(\mathbf{A}) = \prod_{\{i,j\} \in E} P(a_{ij} = 1)^{\frac{\hat{a}_{ij}+1}{2}} (1 - P(a_{ij} = 1))^{1 - \frac{\hat{a}_{ij}+1}{2}}. \qquad (7)$$

Moreover, the 'success probabilities' $P(a_{ij} = 1)$ for node-pairs $\{i, j\}$ are equal to:

$$P(a_{ij} = 1) = \frac{\exp(\sum_{l=1}^{n} f_{ij}^{l} \lambda_l + f_{ij}^{++} \lambda_{++} + f_{ij}^{+-} \lambda_{+-} + f_{ij}^{--} \lambda_{--})}{1 + \exp(\sum_{l=1}^{n} f_{ij}^{l} \lambda_l + f_{ij}^{++} \lambda_{++} + f_{ij}^{+-} \lambda_{+-} + f_{ij}^{--} \lambda_{--})}, \tag{8}$$

where $\lambda_l, \lambda_{++}, \lambda_{+-}$, and λ_{--} 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:

$$L(\lambda_1, \ldots, \lambda_M) = \tag{9}$$
$$\sum_{\{i,j\} \in E} \log(1 + \exp(\sum_{l=1}^{n} f_{ij}^{l} \lambda_l + f_{ij}^{++} \lambda_{++} + f_{ij}^{+-} \lambda_{+-} + f_{ij}^{--} \lambda_{--}))$$
$$- \sum_{l=1}^{n} c_l \lambda_l - c_{++} \lambda_{++} - c_{+-} \lambda_{+-} - c_{--} \lambda_{--}.$$

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 $g : V \to \mathbb{R}^d$ which maps nodes in the network to d-dimensional real-valued vectors. These representations are generally denoted as $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)' \in \mathbb{R}^{n \times d}$, where $\mathbf{x}_i$ 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 $g : V \to \mathbb{R}^d$ to map pairs $\{i, j\} \in E^-$ to more distant representations $\mathbf{x}_i$ and $\mathbf{x}_j$ 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 in unimportant.

CSNE aims to learn the most informative embedding $\mathbf{X}$ for a given signed network $G = (V, E)$ with adjacency matrix $\hat{\mathbf{A}}$. In simple terms, the objective is to find the embedding $\mathbf{X}$ that maximizes the likelihood of observing the signs on the edges in $\hat{\mathbf{A}}$. We formulate this optimization task as a Maximum Likelihood Estimation (MLE) problem, $argmax_{\mathbf{X}} P(\mathbf{A}|\mathbf{X})$. Akin to Kang et al. [17], we do not postulate the likelihood function $P(\mathbf{A}|\mathbf{X})$ directly. Instead, we *do* postulate the density function of the embedding $\mathbf{X}$ conditioned on the signed network, i.e. $p(\mathbf{X}|\mathbf{A})$. Then, we combine $p(\mathbf{X}|\mathbf{A})$ with the MaxEnt prior discussed in Sec. 3.2, $P(\mathbf{A})$, by

means of the Bayes formula. The likelihood function, thus, follows as $P(\mathbf{A}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{A})P(\mathbf{A})}{p(\mathbf{X})}$. 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 $\mathbf{X}$. Effectively, this means that CSNE can a make better use of the embedding space.

The MaxEnt prior $P(\mathbf{A})$ has already been introduced in Sec. 3.2 and its expression given in Eq. 7. For postulating the conditional density $p(\mathbf{X}|\mathbf{A})$, 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. $d_{ij} \triangleq ||\mathbf{x}_i - \mathbf{x}_j||_2$. 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 σ_1 and between pairs $\{i, j\} \in E^-$ from a similar distribution with parameter σ_2, where $\sigma_1 < \sigma_2$. Thus, we have:

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

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

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

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

The resulting likelihood function to optimize is:

$$P(\mathbf{A}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{A})P(\mathbf{A})}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\mathbf{A})P(\mathbf{A})}{\sum_{\mathbf{A}} p(\mathbf{X}|\mathbf{A})P(\mathbf{A})}$$

$$= \prod_{\{i,j\} \in E^+} \frac{\mathcal{N}_+ \left(d_{ij} | \sigma_1^2 \right) P(a_{ij} = 1)}{\mathcal{N}_+ \left(d_{ij} | \sigma_1^2 \right) P(a_{ij} = 1) + \mathcal{N}_+ \left(d_{ij} | \sigma_2^2 \right) (1 - P(a_{ij} = 1))}$$

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

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 $\mathbf{x}_i$ of node i reads:

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

$$+ 2 \sum_{j:\{i,j\} \in E^-} (\mathbf{x}_i - \mathbf{x}_j) P(a_{ij} = 1|\mathbf{X}) \left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2} \right). \tag{14}$$

Intuitively, the first summation in the gradient expression pulls the embedding of each node close to the embeddings of positively

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

Data	Slashdot(a)	Slashdot(b)	Epinions	Wiki-rfa	Bitcoin-α	Bitcoin-otc
$|V|$	77350	82140	131828	11258	3783	5881
$|E|$	468554	500481	711210	171562	14124	21492
$|E|/|V|$	12.11	12.18	10.78	30.47	7.46	7.30
% $|E^+|/|E|$	75%	76%	83%	77%	90%	85%
% Bal. Tri.	85%	86%	89%	73%	83%	85%

connected neighbours. The second summation, on the other hand, pushes this embedding far from those of negatively connected nodes. Finally, after fitting the model, sign predictions can be directly extracted from the posterior probability matrix.

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-α and Bitcoin-otc. 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] uses a deep neural network architecture to learn node embeddings. The objective function optimized by SiNE satisfies the structural balance theory. In this method, nodes are expected to be closer to their friends than their foes. If no negative connections exist for specific nodes, virtual nodes and negative links to these are generated.

- nSNE [34], 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.
- lSNE [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_{test} 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 EvalNE 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, lSNE 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. $\mathbf{x}_{ij} = \mathbf{x}_i \circ \mathbf{x}_j$. In our evaluation, we selected the operators introduced in Grover and Leskovec [11], namely *Average* (($\mathbf{x}_i + \mathbf{x}_j$)/2), *Hadamard* ($\mathbf{x}_i \cdot \mathbf{x}_j$), *Weighted L_1* ($|\mathbf{x}_i - \mathbf{x}_j|$) and *Weighted L_2* ($|\mathbf{x}_i - \mathbf{x}_j|^2$). 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.

Table 2: Sign prediction AUC for all networks. Best performing method per dataset is highlighted in bold.

Methods	Slashdot(a)	Slashdot(b)	Epinions	Wiki-rfa	Bitcoin-α	Bitcoin-otc	Avg. AUC Rank
SiNE	0.850±0.002	0.856±0.002	0.898±0.001	0.816±0.002	0.835±0.012	0.857±0.005	8
nSNE	0.895±0.001	0.894±0.002	0.950±0.001	0.879±0.002	0.810±0.024	0.868±0.010	4.5
lSNE	0.886±0.002	0.893±0.002	0.941±0.002	0.872±0.001	0.854±0.022	0.907±0.005	6
SIGNet	0.887±0.002	0.893±0.002	0.943±0.001	0.848±0.005	0.893±0.002	0.928±0.006	4.5
Prior(pol.)	0.879±0.001	0.885±0.001	0.929±0.001	0.870±0.001	0.858±0.014	0.891±0.002	7
Prior(pol.+tri.)	0.888±0.001	0.894±0.001	0.937±0.001	0.875±0.001	0.874±0.009	0.914±0.004	3
CSNE(pol.)	0.902±0.001	0.908±0.001	0.952±0.001	0.886±0.002	0.896±0.009	0.930±0.002	2
CSNE(pol.+tri.)	**0.904±0.001**	**0.909±0.001**	**0.954±0.000**	**0.887±0.002**	**0.899±0.006**	**0.936±0.002**	1

Hyperparameters. Throughout our evaluation, we set number of dimensions $|d| = 20$ for all methods. For CSNE we fixed the spread parameters $\sigma_1 = 1$ and $\sigma_2 = 2$. We ran SiNE, nSNE, lSNE and CSNE for 500 iterations. This value was empirically found to provide best results in preliminary experiments. Fitting a MaxEnt prior is a convex problem for which we use second order information, therefore, we limited the number of iterations in this case to 20. We performed method hyperparameter tuning on a validation set obtained by further splitting E_{train} in 80% training and 20% validation. The specific method hyperparameters we tuned are as follows. For SiNE, we tuned $\delta = \delta_0 \in \{0.5, 1\}$ and the edge embedding operator. For SIGNet we only tuned the edge embedding operator. For nSNE and lSNE, we varied $\lambda \in \{5e - 5, 2.5e - 5, 1e - 5\}$ and $\beta \in \{0.5, 0.05, 0.005, 0\}$. Finally, for CSNE, including the MaxEnt priors, we did not tune any hyperparameters.

Evaluation measure. We evaluated the methods in terms of Area Under the Curve for the Receiver Operating Characteristic (AUC-ROC). This metric is popular for binary classification tasks and well suited for prediction in the case of class imbalance. Let TP, TN, FP and FN be the elements of a confusion matrix. Then, we can compute the true positive rate as $TPR = \frac{TP}{TP+FN}$ and the false positive rate as $FPR = \frac{FP}{FP+TN}$. The AUC is then the area under the ROC curve created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various thresholds.

4.4 Reproducibility

To guarantee the reproducibility of our experimental evaluation we used the EvalNE toolbox [28]. This Python toolbox aims to simplify and standardize the evaluation of network embedding methods on various downstream tasks. EvalNE uses configuration files that detail the evaluation pipeline. A file describing our sign prediction experiments is available here. This file, together with the datasets and methods reported in Secs. 4.1 and 4.2 allows full reproduction of our experiments.

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.
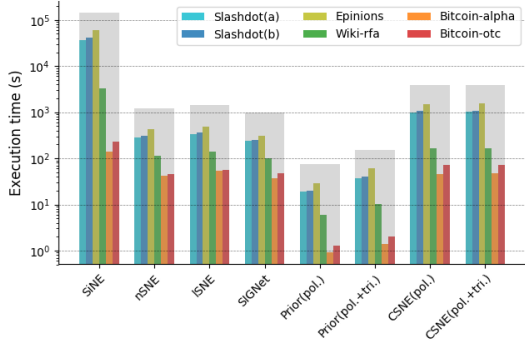
5.1 Sign Prediction

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.

Our results showcase the superior performance of CSNE over the baseline methods on all datasets. The largest difference in AUC to the best performing baseline (2.7%) can be seen for Slashdot(b) while the lowest (0.4%) can be found for Epinions. Additionally, CSNE exhibits a more consistent performance on different networks as compared to other baselines. SIGNet, the best performing baseline on Bitcoin-α, performs significantly worse than CSNE, nSNE and lSNE on Wikipedia. Similarly, nSNE, the best baseline on Slashdot(a), Slashdot(b) and Epinions obtains poor results on Bitcoin-α and Bitcoin-otc.

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

Figure 1: Execution times of all methods for sign prediction on each evaluated network. Gray boxes indicate the total runtime per method on all datasets.
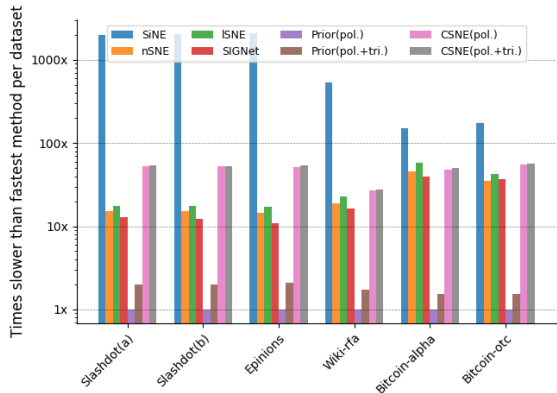

Figure 2: Relative execution times of all methods compared to the fastest approach on each dataset (lower is better).

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 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 σ.
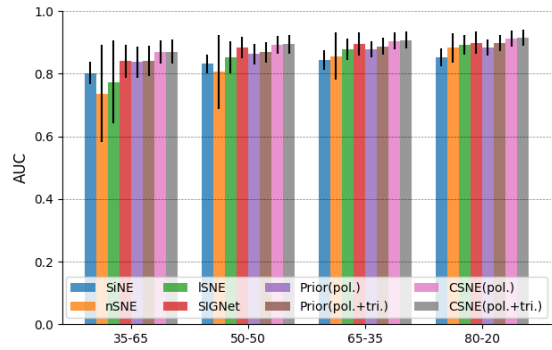
Figure 3: Average sign prediction AUC over all evaluated networks for varying sizes of E_{train} - E_{test}. A black line over each bar denotes the 95% confidence interval for the estimate of the average performance.

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 out 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 σ. The two σ 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
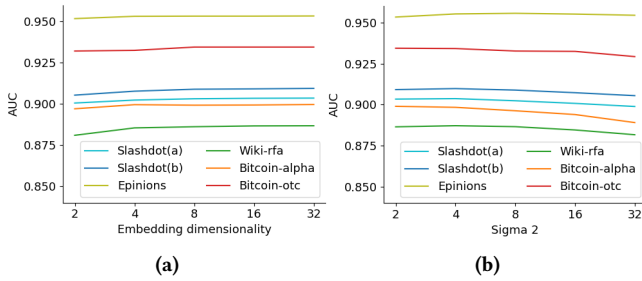
Figure 4: Sign prediction AUC scores of CSNE(pol.+tri.) for (a) different embedding dimensions d and (b) different values of σ_2.
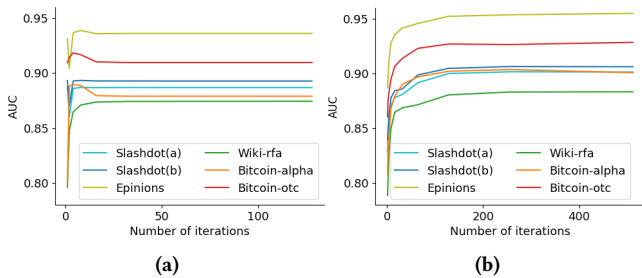

Figure 5: Convergence of (a) the MaxEnt prior and (b) CSNE.

very large values of σ_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.
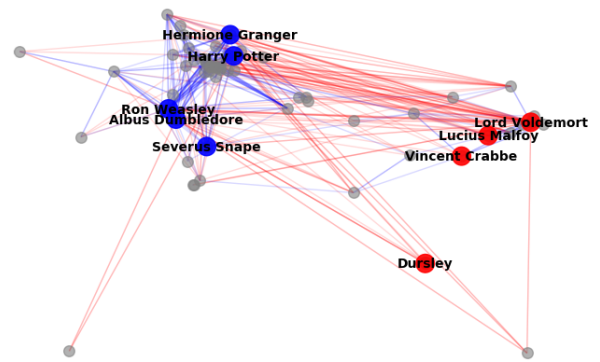

Figure 6: Plot showing the CSNE embeddings of the Harry Potter network. Enemy relationships between book characters are highlighted in red, while friendship is highlighted in blue. A subset of protagonists and antagonists are shown in blue and red, respectively.

We computed a 2-dimensional embedding $\mathbf{X}$ 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 ± 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-partiteness. Another possible line of work is to tackle the extension of CSNE to directed signed networks.

ACKNOWLEDGMENTS

REFERENCES

[1] Florian Adriaens, Alexandru Mara, Jefrey Lijffijt, and Tijl De Bie. 2020. Scalable Dyadic Independence Models with Local and Global Constraints. arXiv:cs.SI/2002.07076
[2] Mikhail Belkin and Partha Niyogi. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proc. of NIPS*. 585–591.
[3] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. 2011. Node classification in social networks. In *Social network data analytics*. Springer, 115–148.
[4] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. 2016. Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the Ninth ACM international conference on Web Search and Data Mining*. ACM, 573–582.
[5] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning graph representations with global structural information. In *Proc. of CIKM*. 891–900.
[6] Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, and Jakub Onufry Wojtaszczyk. 2012. Sitting closer to friends than enemies, revisited. In *International Symposium on Mathematical Foundations of Computer Science*. Springer, 296–307.
[7] Tijl De Bie. 2011. Maximum Entropy Models and Subjective Interestingness: An Application to Tiles in Binary Databases. *Data Min. Knowl. Discov.* 23, 3 (Nov. 2011), 407–446.
[8] Marijtje Duijn, Krista Gile, and Mark Handcock. 2009. A Framework for the Comparison of Maximum Pseudo Likelihood and Maximum Likelihood Estimation of Exponential Family Random Graph Models. *Social Networks* 31 (01 2009), 52–62.
[9] Sheng Gao, Huacan Pang, Patrick Gallinari, Jun Guo, and Nei Kato. 2017. A novel embedding method for information diffusion prediction in social network big data. *IEEE Transactions on Industrial Informatics* 13, 4 (2017), 2097–2105.
[10] Steven Michael Goodreau. 2007. Advances in exponential random graph (p*) models applied to a large social network. *Social networks* 29 2 (2007), 231–248.
[11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proc. of KDD*. 855–864.
[12] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*. ACM, 403–412.
[13] Saket Gurukar, Priyesh Vijayan, Aakash Srinivasan, Goonmeet Bajaj, Chen Cai, Moniba Keymanesh, Saravana Kumar, Pranav Maneriker, Anasua Mitra, Vedang Patel, Balaraman Ravindran, and Srinivasan Parthasarathy. 2019. Network Representation Learning: Consolidation and Renewed Bearing. *ArXiv* abs/1905.00987 (2019).
[14] Fritz Heider. 1946. Attitudes and Cognitive Organization. *The Journal of Psychology* 21, 1 (1946), 107–112. https://doi.org/10.1080/00223980.1946.9917275 arXiv:https://doi.org/10.1080/00223980.1946.9917275 PMID: 21010780.
[15] Mohammad Raihanul Islam, B Aditya Prakash, and Naren Ramakrishnan. 2018. SIGNet: scalable embeddings for signed networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 157–169.
[16] E. T. Jaynes. 1957. Information Theory and Statistical Mechanics. *Phys. Rev.* 106 (May 1957), 620–630. Issue 4. https://doi.org/10.1103/PhysRev.106.620
[17] Bo Kang, Jefrey Lijffijt, and Tijl De Bie. 2019. Conditional Network Embeddings. To appear in *Proc. of ICLR*.
[18] Junghwan Kim, Haekyu Park, Ji-Eun Lee, and U Kang. 2018. Side: representation learning in signed directed networks. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 509–518.
[19] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. 2016. Edge weight prediction in weighted signed networks. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 221–230.
[20] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. 2009. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*. 741–750.
[21] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting Positive and Negative Links in Online Social Networks. In *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, North Carolina, USA) *(WWW '10)*. Association for Computing Machinery, New York, NY, USA, 641–650. https://doi.org/10.1145/1772690.1772756
[22] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1361–1370.
[23] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. 2017. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 577–586.
[24] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58, 7 (2007), 1019–1031.
[25] Chunyu Lu, Pengfei Jiao, Hongtao Liu, Yaping Wang, Hongyan Xu, and Wenjun Wang. 2019. SSNE: Status Signed Network Embedding. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 81–93.
[26] Jie Lu, Junyu Xuan, Guangquan Zhang, and Xiangfeng Luo. 2018. Structural property-aware multilayer network embedding for latent factor analysis. *Pattern Recognit.* 76 (2018), 228–241.
[27] Alexandru Mara, Jefrey Lijffijt, and Tijl De Bie. 2020. Network Representation Learning for Link Prediction: Are we improving upon simple heuristics? arXiv:2002.11522
[28] Alexandru Mara, Jefrey Lijffijt, and Tijl De Bie. 2019. EvalNE: A Framework for Evaluating Network Embeddings on Link Prediction. *arXiv preprint arXiv:1901.09691* (2019).
[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[30] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. 2012. Community detection in social media. *Data Mining and Knowledge Discovery* 24, 3 (2012), 515–554.
[31] Juyong Park and Mark E. J. Newman. 2005. Statistical mechanics of networks. *Physical review. E, Statistical, nonlinear, and soft matter physics* 70 (01 2005), 066117.
[32] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proc. of KDD*. 701–710.
[33] Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290, 5500 (2000), 2323–2326. https://doi.org/10.1126/science.290.5500.2323
[34] Wenzhuo Song, Shengsheng Wang, Bo Yang, You Lu, Xuehua Zhao, and Xueyan Liu. 2018. Learning node and edge embeddings for signed networks. *Neurocomputing* 319 (2018), 42–54.
[35] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *Proc. of WWW*. 1067–1077.
[36] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. 2018. SHINE: Signed Heterogeneous Information Network Embedding for Sentiment Link Prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 592–600. https://doi.org/10.1145/3159652.3159666
[37] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. 2018. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 592–600.
[38] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. 2017. Signed network embedding in social media. In *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 327–335.
[39] Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. 2014. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics* 2 (2014), 297–310.
[40] Shuhan Yuan, Xintao Wu, and Yang Xiang. 2017. SNE: signed network embedding. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 183–195.
[41] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. 2018. Arbitrary-Order Proximity Preserved Network Embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. ACM, New York, NY, USA, 2778–2786. https://doi.org/10.1145/3219819.3219969