Disentangled Link Prediction for Signed Social Networks via Disentangled Representation Learning

Linchuan Xu*, Xiaokai Wei†, Jiannong Cao* and Philip S. Yu‡
*The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
Email: 13901626r@connect.polyu.hk, csjcao@comp.polyu.edu.hk
†Facebook Inc., 1 Hacker Way, Menlo Park, CA, USA
Email: weixiaokai@gmail.com
‡University of Illinois at Chicago, Chicago, Illinois, USA
Email: psyu@uic.edu

***Abstract*—Link prediction is an important and interesting application for social networks because it can infer potential links among network participants. Existing approaches basically work with the homophily principle, i.e., people of similar characteristics tend to befriend each other. In this way, however, they are not suitable for inferring negative links or hostile links, which usually take place among people with different characteristics. Moreover, negative links tend to couple with positive links to form signed networks. In this paper, we thus study the problem of disentangled link prediction (DLP) for signed networks, which includes two separate tasks, i.e., inferring positive links and inferring negative links. Recently, representation learning methods have been proposed to solve the link prediction problem because the entire network structure can be encoded in representations. For the DLP problem, we thus propose to disentangle a node representation into two representations, and use one for positive link prediction and another for negative link prediction. Experiments on three real-world signed networks demonstrate the proposed disentangled representation learning (DRL) method significantly outperforms alternatives in the DLP problem.**

***Index Terms*—link prediction; signed social networks; representation learning;**

I. Introduction

It is common knowledge that human beings are social animals, which explains why people are passionate about participating in social networks. In recent years with highly developed internet and web technologies, it is very convenient for people to reach out and get connected to others. People's involvement in online social networks leaves tremendous social information, which enables data analytics for enhancing user experience in return. Among many ways to enhance user experience, recommending appropriate friends is an effective one, and is widely studied as the link prediction [15] problem, i.e., inferring potential friendships among users. Existing approaches to link prediction basically work with the homophily principle [17] that people of similar characteristics tend to befriend each other. In this way, however, they are not suitable for inferring negative links or hostile links, which usually exist among people with different characteristics.

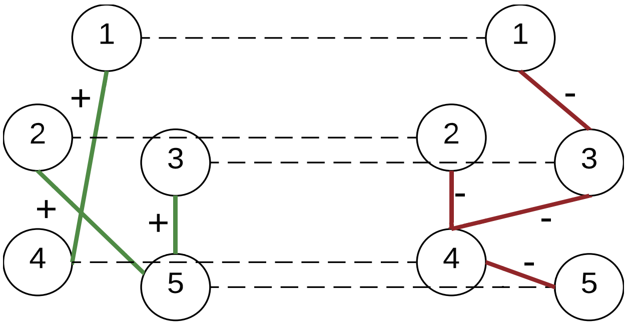


Fig. 1: A disentangled signed network, where green lines denote positive edges while red lines denote negative edges

Negative links are actually as common as positive links due to a very simple fact that human beings are emotional, and have loves and hates. Moreover, negative links tend to couple with positive links to form signed social networks, such as Epinion network with trust and distrust links and Slashdot network with friend and foe links [14]. Hence, the task of inferring negative links are naturally coupled with the task of positive links, which suggests jointly learning for positive link prediction task and negative link prediction task. In this paper, we refer to such problem as **<u>d</u>isentangled <u>l</u>ink <u>p</u>rediction (DLP)** problem for signed networks. The term ”disentangled” emphasizes that the link prediction for a signed network actually consists of two tasks instead of one task. There are two tasks because link prediction is formulated as predicting whether or not a link would come into being by following the conventional setting [15], and there are two type of links.

Recently, representation learning methods have been proposed to solve the link prediction problem [9], [29]. Similarities measured on node representations perform better than topological similarities, such as *Common Neighbor* and *Adamic/Adar* [15], because node representations encode the entire network structure instead of partial information utilized in topological similarities. The basic idea of representation learning in the network scenario is to enforce the similarities of nodes connected by links to be large and those of nodes not connected to be small. However, most existing representation learning models are only designed for networks with exclusively positive links. The problem with the application of them on signed networks is that they would enforce the similarities of nodes with either positive links or negative links to be large.

To address this challenge, we thus propose to disentangle a signed network into a positive sub-network and a negative sub-network as illustrated by Fig. 1, and then learn positive representations from the positive sub-network and negative representations from the negative sub-network. In the rest of the paper, we refer to this idea as *disentangled representation learning (DRL)*.

However, DRL imposes two challenges. Firstly, the idea of existing representation learning models is not suitable for the negative sub-network because it is not natural to enforce nodes negatively connected to be similar. Even if the similarity measured on negative representation is interpreted as negative similarity, another problem comes up. The problem is that nodes showing negative attitudes to a common node, e.g., node 1 and node 4 sharing a common negative neighbor node 3, may also be negatively similar because both of them are enforced to be negatively similar to the common neighbor. But they, e.g., 1 and 4, may have positive relationships according to the theory of weak structure balance [10], [6] that *enemies of my enemies are my friends*. To validate the theory, the number of triads consisting of one positive edge and two negative edges is about 2 times more than the number of triads consisting of three negative links in three real-world signed social networks as presented in Table 1 in the evaluation section.

The second the challenge is how to jointly learn the two representations, which is suggested for the DPL problem. The two representations should be better jointly learned in that although the positive links and negative links are previously disentangled for the purpose of not mixing up opposite relationships, the two types of links are generated in the same signed network. Hence, the joint representation learning can make the representations learned from a single sub-network comprehensive and complete.

To address the first challenge imposed by DRL, we propose to learn negative representations by enforcing nodes connected by negative links to be different instead of being similar. With respect to the second challenge, we propose to jointly learn the two representations by rendering the two representations of the same node somehow similar. The intuition behind this mechanism is that the same node would agree with itself positive attributes exposed in the positive sub-network and negative attitudes exposed in the negative sub-network. To render the two representation similar, each representation would bear some properties of the other one. In this way, each representation may have complete information of the original signed network. We name the process of making each representation have complete information as refinement of representations. The contributions of this paper are summarized as follows:

- To our best knowledge, this is the first attempt to solve the disentangled link prediction problem for signed networks, which requires jointly learning for positive link prediction task and negative link prediction task.
- We propose an idea of disentangled representation learning, which jointly learns two types of representations from a disentangled signed network by rendering the two representations of each node similar.
- We instantiate the proposed DRL in two different models employing different representation refinement mechanisms.
- We conduct experiments on three real-world signed networks to demonstrate that the proposed models significantly outperform alternatives, and show clear evidence why the proposed models have the superior performance.

The rest of the paper is organized as follows. Section 2 presents related work. We develop the proposed models in section 3. In section 4, we present the empirical evaluation. In section 5, we conclude and introduce our future work.

II. Related work

Because of its importance, the link prediction problem has been widely studied in the data mining and machine learning community [15] [19] [16] [1] [3] [28] [29]. The major stream of existing approaches to link prediction basically work with the homophily principle [17], i.e., similarity breeds connection. Previously, similarities are directly measured on network links, such Common Neighbors, Adamic/Adar [15], and Katz [11]. These similarity measurements are effective and generic. However, they only utilize partial information of the network structure.

Recently, methods utilize the entire network structure have been proposed basically by learning low-rank or low-dimensional latent representations. Low-rank representation learning methods usually refer to matrix or tensor factorization [18] [7] while low-dimensional representation learning methods usually refer to network embedding [21] [24] [9] [20] [30] [31]. Matrix or tensor factorization learns representations by approximating the adjacency matrix of network structure while network embedding explicitly encodes the network structure in representations.

Other methods even utilize side information, such as node attributes [12] [8] [3] [27] and auxiliary networks [22] [12], which can provide complementary information to the network structure.

However, none of previous methods are designed for negative link prediction. Moreover, they are not suitable for negative link prediction because the homophily principle are not applicable any more. Although a few studies [25] address a similar problem, they rely on user-generated content, such as posts and comments. In our problem setting, there is only network structure information.

The negative link prediction problem is usually coupled with positive link prediction in signed networks. We thus study the two link prediction problems simultaneously for the first time. Although a few studies utilize negative links in the positive link prediction [23], there are no studies doing the other way around.

III. Methodology

A. Preliminaries

DEFINITION 1. *A* ***signed social network*** *is denoted as*

$G(V, E^+, E^-)$, where V is the set of nodes, E^+ is the set of positive edges, E^- is the set of negative edges, and $E^+ \cap E^- = \emptyset$. Both set of edges can be weighted or unweighted, directed edges, e.g., e_{ij}^+ is the positive edge from node i to node j. If the relationship between nodes has no direction, the edge is replaced by two directed edges e_{ij}^+ and e_{ji}^+. The signed network $G(V, E^+, E^-)$ can be disentangled into two sub-networks, i.e., the positive sub-network $G(V, E^+)$ and the negative sub-network $G(V, E^-)$.

To learn positive representations from the positive sub-network $G(V, E^+)$, the proposed DRL enforces nodes connected by edges to be similar by following the homophily principle, where the similarity is defined as follows:

DEFINITION 2. *The **similarity** of two nodes measured by positive representations is defined as the inner product of two representations normalized by sigmoid function, which can be formulated into the following equation:*

$$p(\boldsymbol{v}_i^+, \boldsymbol{v}_j^+) = \frac{1}{1 + exp\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{v}_j^+\}}, \tag{1}$$

where $\boldsymbol{v}_i^+ \in \mathbb{R}^{D^+}$ and $\boldsymbol{v}_j^+ \in \mathbb{R}^{D^+}$ are positive representations of node i and node j, respectively, and D^+ is the dimension of the positive representation.

To learn negative representations from the negative sub-network $G(V, E^-)$, the proposed DRL enforces nodes connected by edges to be different from each other instead of being similar so as not to violate the theory of weak structure balance as introduced in the introduction. Nevertheless, we can define the similarity measured by negative representations in the same way as that measured by positive representations.

As discussed before, positive representation learning and negative representation learning are jointly performed by rendering the two representations of each node somehow similar, which is to make representations learned from a single sub-network comprehensive and complete. In this paper, two mechanisms are proposed to achieve the refinement of representations.

In the first mechanism, the two representations of the same are enforced to be similar like the proposed positive representation learning idea but with a slightly different similarity definition. The inter-network similarity can be defined similarly to intra-network similarity because each node can be reviewed as implicitly connected to its counterpart in the other sub-network as illustrated in Fig. 1. The implicit edge is established because the same node would agree with itself positive attributes exposed in the positive sub-network and negative attitudes exposed in the negative sub-network. The model working with this mechanism is further named as DRL-C. In the second mechanism, co-regularization is enforced on the two representations of the same node learned from different sub-networks. We further name the model with this mechanism as DRL-R. More details are presented in corresponding sections.

B. The DRL-C Model

Appropriate positive representations should encode the entire network structure. Hence, besides pairs of nodes connected by an edge are enforced to be similar, those not connected are enforced to be different from each other. According to the definition of similarity in Definition 1, the proposed structure preserving mechanism can be achieved by penalizing small similarities of pairs of nodes connected and large similarities of those pairs not connected. We employ the logistic loss to perform the penalty. Hence, the loss function for positive representation learning can be quantified as follows: $l(\boldsymbol{V}^+) =$

$$-\sum_{(i,j)\in E^+} w_{ij}^+ \log(p(\boldsymbol{v}_i^+, \boldsymbol{v}_j^+)) - \sum_{(h,k)\notin E^+} \log(1 - p(\boldsymbol{v}_h^+, \boldsymbol{v}_k^+)) + \lambda ||\boldsymbol{V}^+||_F^2, \tag{2}$$

where $\boldsymbol{V}^+ \in \mathbb{R}^{N \times D^+}$ is the matrix of positive representations, and $N = |V|$, w_{ij}^+ is the weight of the positive edge e_{ij}^+ to reflect the relationship strength, (h, k) is a randomly-sampled pair of nodes not connected, $|| \cdot ||_F^2$ is the square of F_2-norm used as regularization.

The loss function $l(\boldsymbol{V}^-)$ for negative representation learning is similar to $l(\boldsymbol{V}^+)$ expect that pairs of nodes connected by an edge are enforced to be different and those not connected to be similar, where $\boldsymbol{V}^- \in \mathbb{R}^{N \times D^-}$ and D^- is the dimension of negative representations..

To jointly perform positive representation learning and negative representation learning, DRL-C enforces the two representations of the same node to be similar in the existence of the implicit edge between the same node across the positive sub-network and the negative sub-network. Analogous to the similarity defined in Definition 1, the similarity between the representation of the same node can be quantified as follows:

$$p(\boldsymbol{v}_i^+, \boldsymbol{v}_i^-) = \frac{1}{1 + \exp\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{P} \boldsymbol{v}_i^-\}}, \tag{3}$$

where $\boldsymbol{P} \in \mathbb{R}^{D^+ \times D^-}$ is a reconciling matrix. By penalizing small values of $p(\boldsymbol{v}_i^+, \boldsymbol{v}_i^-)$, the positive representation and the negative representation of the same node are enforced to be similar to each other with the help of the reconciling matrix $\boldsymbol{P}$. The reconciling matrix is introduced because positive representations are not directly comparable to negative representations. For the two representations to be similar, the positive representation should bear some information of the negative representation so as to be similar to the negative representation, and vice versa. In this way, DRL-C achieves the task of making representations learned from either sub-network have a complete information of the original signed network.

And similar to embed explicit edges, the logistic loss is employed to perform the penalty for the implicit edges. Hence combining the loss of learning representations from $G(V, E^+)$

and $G(V, E^-)$, the overall loss formulated in the DRL-C model is quantified as follows: $C(\boldsymbol{V}^+, \boldsymbol{P}, \boldsymbol{V}^-) =$

$$\begin{aligned} & l(\boldsymbol{V}^+) - \sum_{i \in V} log(p(\boldsymbol{v}_i^+, \boldsymbol{v}_i^-)) + l(\boldsymbol{V}^-) + \lambda ||\boldsymbol{V}^+||_F^2 \\ & + \beta ||\boldsymbol{P}||_F^2 + \gamma ||\boldsymbol{V}^-||_F^2 \end{aligned} \tag{4}$$

The $L(\boldsymbol{V}^+, \boldsymbol{M}, \boldsymbol{V}^-)$ is not jointly convex on the three variables, i.e., $\boldsymbol{V}^+$, $\boldsymbol{P}$ and $\boldsymbol{V}^-$. But we may solve it alternatingly, and obtain a local optimal. And if the initialization point is appropriately located, the local optimal may performs alike the global optimal.

The derivative w.r.t $\boldsymbol{v}_i^+$ for minimizing Eq. (4) is quantified as follows:

$$\begin{aligned} \frac{\partial C(\boldsymbol{V}^+, \boldsymbol{P}, \boldsymbol{V}^-)}{\partial \boldsymbol{v}_i^+} = & - \sum_{(i,j) \in E^+} \left[\frac{w_{ij}^+ \text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{v}_j^+\}}{1 + \text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{v}_j^+\}} \boldsymbol{v}_j^+ \right] \\ & + \sum_{i \in V} \left[\frac{\text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{P} \boldsymbol{v}_j^-\}}{1 + \text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{P} \boldsymbol{v}_j^-\}} \boldsymbol{P} \boldsymbol{v}_j^- \right] \\ & - \sum_{(i,k) \notin E^+} \left[\frac{\boldsymbol{v}_k^+}{1 + \text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{v}_k^+\}} \right] \\ & + 2\lambda \boldsymbol{v}_i^+, \end{aligned} \tag{5}$$

The derivative w.r.t $\boldsymbol{v}_i^-$ is quantified as follows:

$$\begin{aligned} \frac{\partial C(\boldsymbol{V}^+, \boldsymbol{P}, \boldsymbol{V}^-)}{\partial \boldsymbol{v}_i^-} = & - \sum_{(i,j) \in E^-} \left[\frac{w_{ij}^- \text{exp}\{-(\boldsymbol{v}_i^-)^\top \boldsymbol{v}_j^-\}}{1 + \text{exp}\{-(\boldsymbol{v}_i^-)^\top \boldsymbol{v}_j^-\}} \boldsymbol{v}_j^- \right] \\ & - \sum_{i \in V} \left[\frac{\text{exp}\{-(\boldsymbol{v}_j^+)^\top \boldsymbol{P} \boldsymbol{v}_i^-\}}{1 + \text{exp}\{-(\boldsymbol{v}_j^+)^\top \boldsymbol{P} \boldsymbol{v}_i^-\}} \boldsymbol{P} \boldsymbol{v}_i^+ \right] \\ & + \sum_{(i,k) \notin E^-} \left[\frac{\boldsymbol{v}_k^-}{1 + \text{exp}\{-(\boldsymbol{v}_i^-)^\top \boldsymbol{v}_k^-\}} \right] \\ & + 2\gamma \boldsymbol{v}_i^-, \end{aligned} \tag{6}$$

The derivative w.r.t $\boldsymbol{P}$ is quantified as follows:

$$\begin{aligned} \frac{\partial C(\boldsymbol{V}^+, \boldsymbol{P}, \boldsymbol{V}^-)}{\partial \boldsymbol{P}} = & - \sum_{i \in V} \left[\frac{\text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{P} \boldsymbol{v}_j^-\}}{1 + \text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{P} \boldsymbol{v}_j^-\}} \boldsymbol{v}_i^+ (\boldsymbol{v}_j^-)^\top \right] \\ & + 2\beta \boldsymbol{P}, \end{aligned} \tag{7}$$

C. The DRL-R Model

DRL-R is similar to DRL-C except for the refinement mechanism. Specifically, DRL-R enforces a co-regularization on the two representations of the same node, where the co-regularization can be achieved by employing a reconciling matrix as follows:

$$\min_{\boldsymbol{V}^+, \boldsymbol{V}^-} ||\boldsymbol{V}^+ \boldsymbol{M} - \boldsymbol{V}^-||_F^2, \tag{8}$$

where $\boldsymbol{M} \in \mathbb{R}^{D^+ \times D^-}$. The intuition behind the co-regularization is that the difference between the positive representation and the negative representation of the same node should be minimized with help of the reconciling matrix $\boldsymbol{M}$. To minimize the difference, the positive representation should bear some information of the negative representation, and vice versa. In this way, both positive representations and negative representations can be more comprehensive and complete. With the reconciling matrix, the two representations do not need to be exactly the same. And the two representations should better not be the same because they are learned from two networks with edges different in nature.

Combining the loss of learning representations from the network structure and the co-regularization loss, the overall loss can be quantified as follows:

$$\begin{aligned} L(\boldsymbol{V}^+, \boldsymbol{M}, \boldsymbol{V}^-) = & l(\boldsymbol{V}^+) + ||\boldsymbol{V}^+ \boldsymbol{M} - \boldsymbol{V}^-||_F^2 + l(\boldsymbol{V}^-) \\ & + \lambda ||\boldsymbol{V}^+||_F^2 + \beta ||\boldsymbol{M}||_F^2 + \gamma ||\boldsymbol{V}^-||_F^2, \end{aligned} \tag{9}$$

where λ, β, and $\gamma \in \mathbb{R}$ are regularization coefficients.

The derivative w.r.t $\boldsymbol{v}_i^+$ for minimizing Eq. (9) is quantified as follows:

$$\begin{aligned} \frac{\partial L(\boldsymbol{V}^+, \boldsymbol{M}, \boldsymbol{V}^-)}{\partial \boldsymbol{v}_i^+} = & - \sum_{(i,j) \in E^+} \left[\frac{w_{ij}^+ \text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{v}_j^+\}}{1 + \text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{v}_j^+\}} \boldsymbol{v}_j^+ \right] \\ & + \sum_{(i,k) \notin E^+} \left[\frac{\boldsymbol{v}_k^+}{1 + \text{exp}\{-(\boldsymbol{v}_i^+)^\top \boldsymbol{v}_k^+\}} \right] \\ & + 2 \left[(\boldsymbol{v}_i^+)^T \boldsymbol{M} - (\boldsymbol{v}_i^-)^T \right] \boldsymbol{M}^T + 2\lambda (\boldsymbol{v}_i^+), \end{aligned} \tag{10}$$

The derivative w.r.t $\boldsymbol{v}_i^-$ is quantified as follows:

$$\begin{aligned} \frac{\partial L(\boldsymbol{V}^+, \boldsymbol{M}, \boldsymbol{V}^-)}{\partial \boldsymbol{v}_i^-} = & - \sum_{(i,j) \in E^-} \left[\frac{w_{ij}^- \text{exp}\{-(\boldsymbol{v}_i^-)^\top \boldsymbol{v}_j^-\}}{1 + \text{exp}\{-(\boldsymbol{v}_i^-)^\top \boldsymbol{v}_j^-\}} \boldsymbol{v}_j^- \right] \\ & + \sum_{(i,k) \notin E^-} \left[\frac{\boldsymbol{v}_k^-}{1 + \text{exp}\{-(\boldsymbol{v}_i^-)^\top \boldsymbol{v}_k^-\}} \right] \\ & + 2 \left[(\boldsymbol{v}_i^-)^T - (\boldsymbol{v}_i^+)^T \boldsymbol{M} \right] + 2\lambda (\boldsymbol{v}_i^-), \end{aligned} \tag{11}$$

For the reconciling matrix $\boldsymbol{M}$, the problem turns into solving the following minimization:

$$\min_{\boldsymbol{M}} ||\boldsymbol{V}^+ \boldsymbol{M} - \boldsymbol{V}^-||_F^2 + \beta ||\boldsymbol{M}||_F^2, \tag{12}$$

which is convex. Hence, the optimal $\boldsymbol{M}$ can be obtained by setting the derivative of Eq. (7) w.r.t $\boldsymbol{M}$ to zero. After easy calculations, the optimal $\boldsymbol{M}$ is as follows:

$$\boldsymbol{M} = \left[(\boldsymbol{V}^+)^T \boldsymbol{V}^+ + \beta I \right]^{-1} (\boldsymbol{V}^+)^T \boldsymbol{V}^-, \tag{13}$$

where $\boldsymbol{I} \in \mathbb{R}^{D^+ \times D^+}$ is an identity matrix

D. The Optimization Algorithm

The optimization problem of both the DRL-C model and the DRL-R model is a joint minimization problem over three variables, i.e., positive representation, reconciling matrix, and negative representation. Hence, we replace it with a sequence of easier sub-problems by an alternating optimization algorithm [5]. More specifically, the joint minimization problem is alternatingly solved with respect of one of the three variables at a time with other variables fixed. For each variable, we then solve it according to the corresponding problem. In our case, the sub-problems of DRL-C with respect to all the variables can be solved by gradient-based algorithms, e.g.,

Algorithm 1: The optimization algorithm

```
Input  : $G(V, E^+, E^-)$, $D^+$, $D^-$, negative ratio, $\lambda$, $\beta$, $\gamma$
Output: $\boldsymbol{V}^+$, $\boldsymbol{V}^-$
Pre-training $\boldsymbol{V}^+$ and $\boldsymbol{V}^-$;
while (not converge) do
    Fixing $\boldsymbol{V}^+$ and $\boldsymbol{V}^-$, find the optimal $\boldsymbol{M}$ with Eq. (8), or find the optimal $\boldsymbol{P}$ with gradient descent;
    Fixing $\boldsymbol{V}^-$, and $\boldsymbol{M}$ or $\boldsymbol{P}$, find the optimal $\boldsymbol{V}^+$ with gradient descent;
    Fixing $\boldsymbol{V}^+$, and $\boldsymbol{M}$ or $\boldsymbol{P}$, find the optimal $\boldsymbol{V}^-$ with gradient descent;
return $\boldsymbol{V}^+$, $\boldsymbol{V}^-$
```

steepest descent or L-BFGS. The updating rule for positive representation is illustrated as follows:

$$(\boldsymbol{v}_i^+)^{p+1} = (\boldsymbol{v}_i^+)^p - d^{p+1} \times \frac{\partial C(\boldsymbol{V}^+, \boldsymbol{P}, \boldsymbol{V}^-)}{\partial \boldsymbol{v}_i^+}, \tag{14}$$

where p denotes the p-th iteration, $d \in \mathbb{R}$ is the descent rate. Updating rules for other variables are similar and thus omitted. For all gradient descent algorithms, the descent rate in each iteration is obtained by backtracking line search [2].

The sub-problems of DRL-R with respect to positive representation and negative representation can be solved similarly to those of DRL-C. But the sub-problem w.r.t the reconciling matrix $\boldsymbol{M}$ can be directly solved by Eq. (13). The optimization algorithm for solving the DRL-C model and the DRL-R model is similar, and we thus present a unified one for both of them. The psuedo-codes of the proposed alternating optimization algorithm are presented in Algorithm 1.

The optimization algorithm starts with pre-training $\boldsymbol{V}^+$ and $\boldsymbol{V}^-$, which is performed to assign appropriate initialization values to $\boldsymbol{V}^+$ and $\boldsymbol{V}^-$ so as to obtain a good local optimal. The details of pre-training are presented in the following paragraphs. The parameter negative ratio is the ratio of the number of e_{hk}^+ to that of e_{ij}^+, and the number of e_{hk}^- to that of e_{ij}^-. The negative ratio is introduced because the number of pairs of nodes not connected by an edge grows at a quadratic speed when the number of nodes grows. And hence it may not be efficient to apply the proposed algorithm to large-scale networks. To compromise, the number of e_{hk}^+ is set to several times larger than that of e_{ij}^+. It is observed that the algorithm works well with this parameter in the experiments.

	Epinions	Slashdot	Wikipedia
Type	unweighted, directed		
# Nodes	19,714	30,556	5,569
# Edges	632,072	460,812	168,060
# + - - triads	446,728	73,667	109,052
# - - - triads	156,243	30,447	49,893
# + + - triads	436,244	57,609	248,515
# + + + triads	13,159,091	1,391,154	1,969,394

TABLE I: Network statistics

Pre-training is an important part of an optimization algorithm as it can initialize a model to a point in parameter space that renders the learning process more effective [4]. In our case, to make the learning process more effective, $\boldsymbol{V}^+$ should be pre-trained to take values learned from the positive subnetwork, and $\boldsymbol{V}^-$ should take values learned from the negative sub-network. Representation learning from the positive subnetwork has been formulated into an optimization problem that minimizes the loss indicated by $l(\boldsymbol{V}^+)$ in Eq. (2). Similarly, we solve it by gradient descent. The derivative is computed as follows:

$$\begin{aligned}\frac{\partial l(\boldsymbol{V}^+)}{\partial \boldsymbol{v}_i^+} = &- \sum_{(i,j)\in E^+} \left[\frac{w_{ij}^+ \exp\{-(v_i^+)^\top \boldsymbol{v}_j^+\}}{1+\exp\{-(v_i^+)^\top \boldsymbol{v}_j^+\}} \times \boldsymbol{v}_j^+ \right] \\ &+ \sum_{(h,k)\notin E^+} \left[\frac{\boldsymbol{u}_k}{1+\exp\{-(v_i^+)^\top \boldsymbol{v}_k^+\}} \right] + 2\lambda(\boldsymbol{v}_i^+)\end{aligned} \tag{15}$$

Algorithm 1 is essentially a block-wise coordinate descent algorithm [26] with positive representations and negative representations as block variables. So convergence can be guaranteed based on the general proof of convergence for block-wise coordinate descent. Moreover, we observe that Algorithm 1 converges very fast in terms of the outer iterations in our experiments, which is presented in the evaluation section.

IV. Empirical Evaluation

A. Datasets

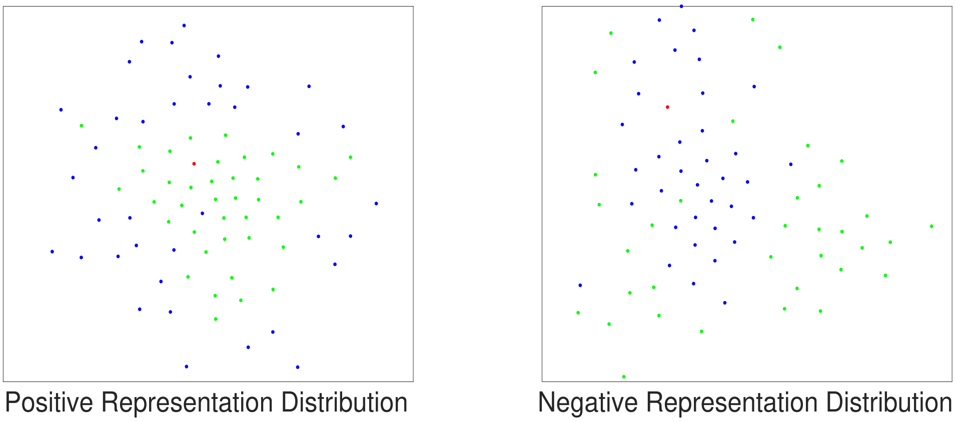
We study three real-world signed social networks [14], [13] as follows:

- Epinions network is a trust and dis-trust network among users of the Epinions product review Web site. In our experiments, we filter out inactive users by setting the frequency of appearance in edges as 10.
- Slashdot network is a like and dis-like network among users of the blog Slashdot social network. The dataset is sampled up to the time of Feb. 21, 2009. Similar to the Epinions network, the node frequency is set as 5.
- The Wikipedia voting network is a for-against network, where a signed link indicates a positive or negative vote by one user on the promotion to admin status of another. In experiments, the node frequency is set as 5.

Statistics for all the three datasets are presented in Table 1.

B. Implementation Settings

For all the representation learning models, the dimension of representations is set as 128 which are commonly used in previous studies [21], [24], [9]. For the implementation of Algorithm 1, negative ratio is set as 5 as used in LINE [24], all the regularization coefficients are set as 1, commonly used settings are used in backtracking line search [2], and the relative loss that determines whether the gradient-based descent algorithm converges is set as 0.001.


Positive Representation Distribution

Negative Representation Distribution

Fig. 2: Visualization of selected nodes from Wikipedia network, where nodes in red are expected to be the centroids in each figure because all the other nodes are selected according to whether they are connected to the red node or not. Specifically, nodes in green are connected to the red node while nodes in blue are not.

C. Baselines

The proposed models are evaluated against one heuristic similarity metric, one matrix factorizaiton model, and three node representation learning models, which are listed as follows:

- TNS [23]: Transitive Node Similarity (TNS) is calculated by the product of Jaccard Coefficient based similarities between two nodes. Moreover, it can be modified to take negative links into account. Other similarity metrics, such as using Common Neighbors and Adamic/Adar, are omitted in the comparison because they significantly underperform node2vec as suggested in its paper [9].
- Matrix Factorization (MF) [18]: The model learns latent features from the topological structure of a graph by performing a matrix completion task.
- DeepWalk [21]: This model learns node representations that encode structural information by using truncated random walk as input.
- LINE [24]: This model learns node representations by preserving first-order structural information or second-order structural information.
- node2vec [9]: The representation learning process of this model is similar to DeepWalk. But it employs a more flexible definition of neighborhood to facilitate random walks.

D. Experiment Settings

Because all the baseline representations models are only applicable to networks with one type of links, we thus apply them to $G(V, E^+)$ and $G(V, E^-)$ to learn positive representations and negative representations, respectively. In this way, both $G(V, E^+)$ and $G(V, E^-)$ are treated as unsigned networks. For the proposed DRL-C and DRL-R, positive representations and negative representations are jointly learned as presented before. Similarities measured on positive representations, i.e., positive similarities, are employed to infer positive links while negative similarities are employed to infer negative links. All the similarities are computed in Definition 1. Although some baselines may not explicitly optimized for the defined similarity, the inference of new links are performed based on ranking similarities of candidate pairs of nodes. Thus, AUC (ares under the curve), which is ranking-based, is employed as the evaluation metric. For Transitive Node Similarity (TNS), negative links are leveraged for positive link prediction while only negative links are only utilized for the negative link prediction because it does not specify how to incorporate positive links for negative link prediction.

Another important setting is about how to perform negative link prediction. Because all baselines are deigned based on the homophily principle, the inference of negative links still rank pairs of node with larger similarities in front of those with smaller similarities. However, the proposed models enforce nodes connected by negative links to be different. Hence, the inference of negative links using representations learned by the proposed models is done in the opposite way.

E. Case Study

Before evaluating the link prediction performance, this section evaluates whether the proposed representation learning methods work as expected. If they indeed work, nodes connected by positive edges should be more similar than those not connected by positive edges when the similarity is measured by positive representations. Moreover, nodes connected by negative edges should be less similar than those not connected by negative edges when the similarity is measured by negative representations. To see whether these two points hold, we select one node with user ID Triona from Wikipedia network and visualize it together with nodes positively connected to it, and select another node with user ID Aitias from Wikipedia network and visualize it together with nodes negatively connected to it. Moreover, the same number

		Positive Link Prediction				Negative Link Prediction			
Dataset	Model	20%	40%	60%	80%	20%	40%	60%	80%
Wikipedia	TNS	53.87	56.34	58.78	60.01	43.33	45.32	46.69	49.05
	MF	61.35	65.66	68.89	71.76	50.16	53.05	55.08	55.89
	DeepWalk	66.63	71.94	77.88	79.35	61.78	64.12	67.23	68.89
	LINE(1st)	55.03	64.57	70.25	74.05	38.88	44.23	48.11	51.97
	LINE(2nd)	61.70	68.66	71.27	72.62	51.63	55.20	55.88	57.46
	node2vec	67.67	70.46	76.36	78.02	62.15	64.66	67.14	68.63
	DRL-C	81.21	90.67	91.92	91.88	76.00	**80.77**	83.35	85.28
	DRL-R	**84.87**	**92.10**	**92.99**	**93.53**	**76.61**	79.14	**87.18**	**88.56**
Epinion	TNS	65.18	69.89	73.66	75.18	47.67	49.86	52.13	53.36
	MF	75.55	79.02	82.87	85.16	56.69	59.02	62.67	64.12
	DeepWalk	83.36	87.93	89.82	91.21	65.75	69.75	72.23	73.92
	LINE(1st)	67.89	75.52	79.91	82.66	33.53	36.01	38.11	39.69
	LINE(2nd)	82.12	87.24	88.75	89.76	58.13	59.94	61.25	63.00
	node2vec	82.45	86.92	88.99	90.02	66.24	71.23	71.92	72.95
	DRL-C	84.52	90.46	**92.68**	92.99	**88.90**	91.87	93.19	93.89
	DRL-R	**84.93**	**90.69**	91.50	**93.90**	88.14	**93.85**	**94.85**	**95.05**
Slashdot	TNS	54.32	56.24	58.10	58.91	46.66	49.32	51.56	53.10
	MF	63.34	65.62	67.15	68.51	53.35	55.87	57.72	58.36
	DeepWalk	68.10	71.11	73.23	74.56	63.17	69.13	71.32	75.38
	LINE(1st)	47.28	51.41	54.45	57.74	34.96	35.43	39.86	41.73
	LINE(2nd)	65.70	68.73	69.94	70.54	56.92	60.97	62.94	63.94
	node2vec	67.08	70.61	72.85	73.13	65.16	67.61	71.15	72.10
	DRL-C	70.89	73.69	72.78	81.10	79.80	88.61	90.32	90.97
	DRL-R	**71.20**	**74.18**	**74.81**	**82.28**	**81.67**	**89.40**	**91.42**	**92.65**

TABLE II: AUC scores on link prediction when different percentages of links are used in the training phase.

of nodes not connected to each node are randomly sampled and visualized. The results are presented in Fig. 2.

We can see from positive representation distribution that nodes with positive links to the selected node are closer to the selected node than those nodes not connected. When two nodes are closer, the similarity of them is larger. Similarly, in negative representation distribution, nodes with negative links to the selected node are far away from the selected node while those nodes not connected are close to the selected node. Hence, the proposed representations learning methods work as expected.

F. Prediction Performance Comparison

Table 2 presents AUC scores of 4 runs of experiments where different percentage of links are used in the training process, and remaining ones are used as test links. For the evaluation purpose, the same number of non-existing links are randomly sampled in each task. All the scores have been multiplied by 100%.

Table 2 shows that the proposed models consistently outperform all baselines on all datasets, and the advantage is significant on negative link prediction. Generally, the reason behind the superior performance of the proposed models over all baselines is that the proposed models can utilize the complete information of the signed networks while existing baselines can only utilize partial information, i.e., either positive links or negative links. This is also the reason that all representations learning methods outperform TNS at most times. The reason why network embedding models outperform matrix factorization is that network embedding explicitly preserve the entire network structure in representations while matrix factorization only approximates the existing edges using representations.

When it comes to reasons why the complete information is important, we put as follows. For positive link prediction, the importance of information leveraged from the negative subnetwork is reflected in two folds. Firstly, nodes connected by negative links would have small similarities enforced by the negative representation learning, which can be transferred to positive representations because positive representations are made similar to negative representations. Secondly, nodes not connected by negative links would have relatively large similarities, and can be transferred to positive representations as well. Both the small similarities of nodes connected by negative links and relatively larger similarities of nodes not connected by negative links are demonstrated in case study. The case with negative link prediction is similar.

The poor performance of all the baselines on the negative link prediction may be explained by the problem discussed in the introduction, i.e., two nodes sharing a common negative neighbor may also be negatively close because both of them are presented to be negatively close to the common node, but they may have positive relationships according to the theory of weak structure balance [10], [6]. To visually demonstrate this point, we present the distribution of negative closeness between the positively connected nodes in + - - triads, and between nodes of test negative links in Fig. 3. The results are obtained from the Wikipedia network when 90% edges are used as training data. Similar results from other two networks are omitted. In Fig.3, the negative closeness of positive connected nodes tend to be larger than that of negatively connected nodes produced by all the baselines. As a result, the ranking-based AUC scores cannot be high. But the proposed models can address this problem by presenting nodes negatively connected to be apart. In this way, the negative

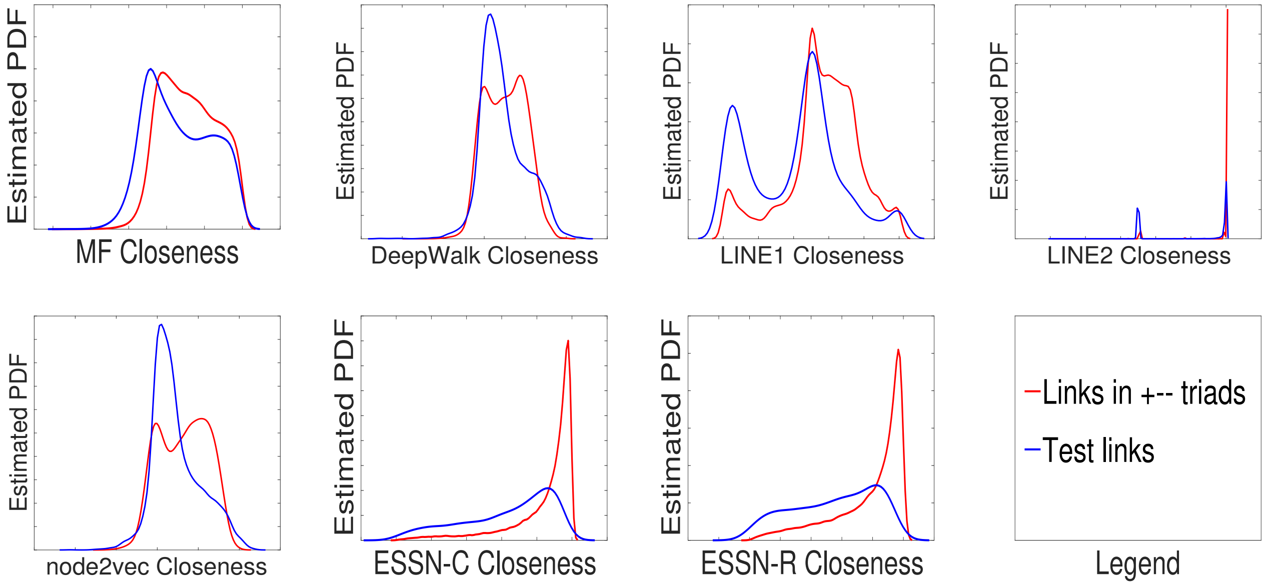


Fig. 3: Distribution of negative closeness between the positively connected nodes in + - - triads, and between nodes of test negative links, which are denoted in red line and blue line, respectively.

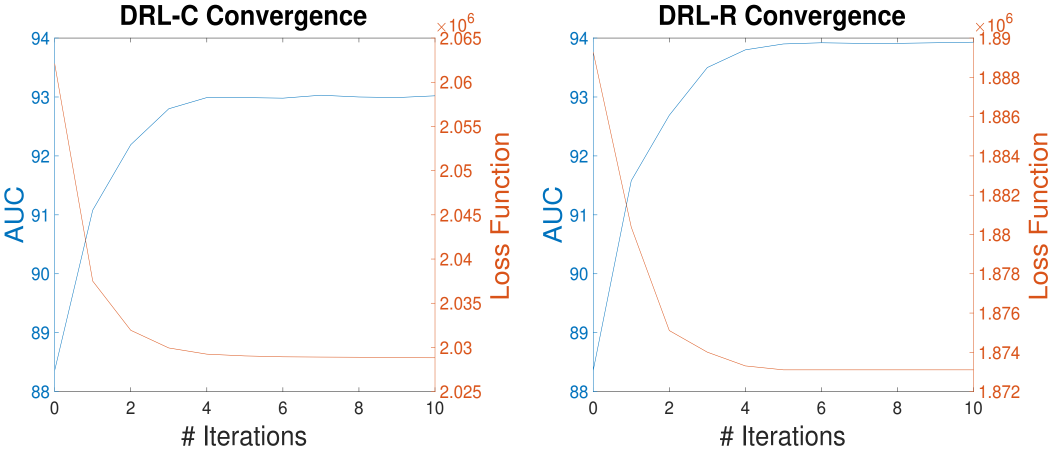


Fig. 4: Convergence analysis

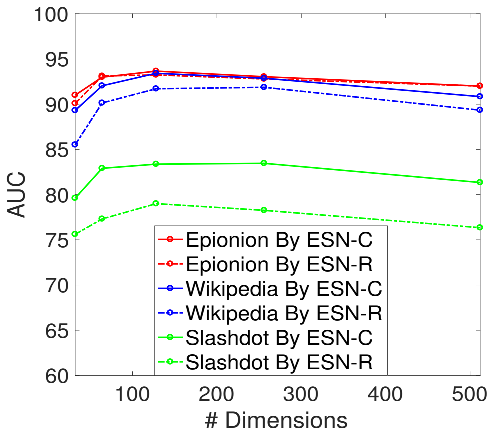


Fig. 5: Parameter sensitivity

closeness of positive connected nodes is expected to be larger than that of negatively connected nodes, which is indeed the case shown in Fig. 3.

G. Convergence Analysis

This section studies the convergence of the proposed alternating algorithm as indicated in Algorithm 1. Specifically, we study the performance of the algorithm on applications with respect to the number of outer iterations. We only present the performance on the link prediction task for Epinion when 80% of links used in the training phase in Fig. 4 because other experiments show similar results. It shows that the algorithm converges very fast and can usually converge to stable performance after about 5 iterations.

H. Parameter Sensitivity

This section evaluates the performance w.r.t the dimension in the positive link prediction tasks where 90% of links used as training data. The experiment results are presented in Fig. 5. It is shown that the performance of the proposed models on the two tasks is not much sensitive to the dimension of representations as long as the dimension is not too small (e.g., 32) or too large (e.g., 512).

V. Conclusion and future work

This paper proposes two models, i.e., DRL-C and DRL-R, to learn representations for disentangled link prediction problem. Both the two models instantiate the idea of disentangled representation learning, i.e., disentangling a signed network into a positive sub-network and a negative sub-network, and then jointly learning positive representations and negative representations. The difference between the two models is the mechanism for refining the positive representation and the negative representation of the same node. The two representations should better be jointly learned because the positive sub-network and the negative sub-network are originally generated in the same signed network, and hence joint representation learning can render representations have complete information about the original network. Moreover, positive links have inseparable relationships with negative links as suggested by the theory of weak structural balance. Experiments on three real-world signed social networks demonstrate that the two models significantly outperform other representation learning models that can only learn representations from unsigned networks. In the future, we will explore other mechanisms for jointly learning representations from the two sub-networks of a signed network.

Acknowledgment

The work described in this paper was partially supported by the funding for Project of Strategic Importance provided by The Hong Kong Polytechnic University (Project Code : 1-ZE26), the funding for demonstration project on large data provided by The Hong Kong Polytechnic University (project account code : 9A5V), and NSF through grants IIS-1526499, and CNS-1626432, and NSFC 61672313.

References

[1] Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer, 2011.

[2] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[3] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Who to follow and why: link prediction with explanations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1266–1275. ACM, 2014.

[4] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[5] James C Bezdek and Richard J Hathaway. Some notes on alternating optimization. In *AFSS International Conference on Fuzzy Systems*, pages 288–300. Springer, 2002.

[6] James A Davis. Clustering and structural balance in graphs. *Social networks. A developing paradigm*, pages 27–34, 1977.

[7] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):10, 2011.

[8] Sheng Gao, Ludovic Denoyer, and Patrick Gallinari. Temporal link prediction by integrating content and structure information. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1169–1174. ACM, 2011.

[9] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[10] Fritz Heider. Attitudes and cognitive organization. *The Journal of psychology*, 21(1):107–112, 1946.

[11] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[12] Vincent Leroy, B Barla Cambazoglu, and Francesco Bonchi. Cold start link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 393–402. ACM, 2010.

[13] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650. ACM, 2010.

[14] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370. ACM, 2010.

[15] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[16] Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 243–252. ACM, 2010.

[17] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[18] Aditya Menon and Charles Elkan. Link prediction via matrix factorization. *Machine Learning and Knowledge Discovery in Databases*, pages 437–452, 2011.

[19] Kurt Miller, Michael I Jordan, and Thomas L Griffiths. Nonparametric latent feature models for link prediction. In *Advances in neural information processing systems*, pages 1276–1284, 2009.

[20] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proc. of ACM SIGKDD*, pages 1105–1114, 2016.

[21] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[22] Yizhou Sun, Jiawei Han, Charu C Aggarwal, and Nitesh V Chawla. When will it happen?: relationship prediction in heterogeneous information networks. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 663–672. ACM, 2012.

[23] Panagiotis Symeonidis, Eleftherios Tiakas, and Yannis Manolopoulos. Transitive node similarity for link prediction in social networks with

positive and negative links. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 183–190. ACM, 2010.

[24] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. ACM, 2015.

[25] Jiliang Tang, Shiyu Chang, Charu Aggarwal, and Huan Liu. Negative link prediction in social media. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 87–96. ACM, 2015.

[26] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[27] Xiaokai Wei, Bokai Cao, and Philip S. Yu. Unsupervised feature selection on networks: A generative view. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2215–2221, 2016.

[28] Xiaokai Wei, Sihong Xie, and Philip S. Yu. Efficient partial order preserving unsupervised feature selection on networks. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 82–90, 2015.

[29] Xiaokai Wei, Linchuan Xu, Bokai Cao, and Philip S Yu. Cross view link prediction by learning noise-resilient representation consensus. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1611–1619. International World Wide Web Conferences Steering Committee, 2017.

[30] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S. Yu. Embedding identity and interest for social networks. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, pages 859–860, 2017.

[31] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S. Yu. Embedding of embedding (EOE): joint embedding for coupled heterogeneous networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, pages 741–749, 2017.