Cross View Link Prediction by Learning Noise-resilient Representation Consensus

Xiaokai Wei*, Linchuan Xu†, Bokai Cao* and Philip S. Yu*
* Department of Computer Science, University of Illinois at Chicago,
{xwei2, caobokai, psyu}@uic.edu
† Department of Computing, The Hong Kong Polytechnic University,
cslcxu@comp.polyu.edu.hk

ABSTRACT

Link Prediction has been an important task for social and information networks. Existing approaches usually assume the completeness of network structure. However, in many real-world networks, the links and node attributes can usually be partially observable. In this paper, we study the problem of **Cross View Link Prediction (CVLP)** on partially observable networks, where the focus is to recommend nodes with only links to nodes with only attributes (or vice versa). We aim to bridge the information gap by learning a robust consensus for link-based and attribute-based representations so that nodes become comparable in the latent space. Also, the link-based and attribute-based representations can lend strength to each other via this consensus learning. Moreover, attribute selection is performed jointly with the representation learning to alleviate the effect of noisy high-dimensional attributes. We present two instantiations of this framework with different loss functions and develop an alternating optimization framework to solve the problem. Experimental results on four real-world datasets show the proposed algorithm outperforms the baseline methods significantly for cross-view link prediction.

1. INTRODUCTION

In the past decade, there have been an increasing number of information networks from a wide range of domains. Study on computer networks, biological and social networks has attracted great attention from the research community [7] [4] [26]. Link prediction [1, 2], which aims at recommending potential links between network nodes, is an important step to understand and study the characteristics of these networks. For instance, in bioinformatics, by predicting protein interaction links, one does not need to conduct expensive experiments on all possible pairs and can spend the resource wisely on the most likely interaction. For social media websites, such as Facebook and Twitter, it is fundamental to grow the user base and enhance user engagement with link prediction techniques. For security analysts/agencies, predicting (currently unobserved) links can reveal hidden but important relationship among terrorists and provides additional insights for understanding organizational structures of terrorist-attack activities.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052575

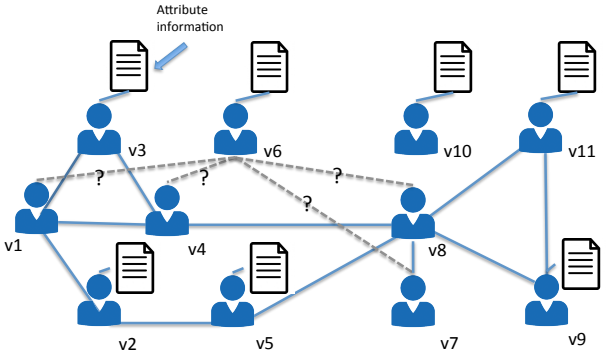


Figure 1: An example of networks with partially observable links and attributes

Many methods have been proposed for the task of link prediction [11, 1, 9, 2]. However, in various social and information networks, it is common that certain nodes do not have any link information revealed [25] and make these methods not applicable:

- In real-world social networks (e.g., Twitter, Facebook and LinkedIn), link prediction for new users usually has the challenge of cold start problem, since these users do not have any connection. Besides, some users may choose a strict privacy setting that restricts the visibility of their connections, personal information or posts[1][2]. Recommending links in such a partially observable setting could enhance user experience.
- In bioinformatics information networks, for example, studying protein interaction could help researchers better understand many biological processes. However, it is infeasible to collect all the experimental data for all the possible pairs of protein.
- In terrorist-attack networks, nodes represent terrorist activities and links represent terrorist attacks in which the same terrorist group is involved. Detecting hidden links in these networks is useful for understanding the underlying structure of terrorist-attack activities. However, the complete linkages between attacks are highly difficult to resolve [13].

Nonetheless, nodes in many social/information networks are often equipped with features/attributes, such as user attributes in so-

[1] https://www.facebook.com/help/325807937506242/
[2] https://help.linkedin.com/app/answers/detail/a_id/52

cial networks, paper content in co-authorship networks and gene properties in biological networks. These node attributes can help when the link information is not observable. For example, for a new user joining a social network with few friendship/following connections (i.e., links), we can leverage his/her user profile (i.e., node attributes) filled out in the registration process to suggest potential links to such a new user, based on the profile similarity.

However, due to the difficulty in data collection, the node attributes of real-world networks also tend to be partially observable in a variety of scenarios and this poses additional challenges for link prediction.

- In online social networks, some users might not fill up profile information when registering or have not yet started to write posts. Besides, a user might choose a privacy level with which no one or only friends could view his/her posts and profile information.
- For information networks in domain of bioinformatics, it can be costly to obtain features for certain genes or proteins.
- In terrorist network, the difficulty of collecting attributes/profiles for different terrorists varies. For example, the information of terrorists with higher ranks is often protected better than that of an ordinary terrorist. Also, it is usually difficult to obtain all the necessary attributes for a newly joined terrorist.

Hence, for real-world networks, assuming *partially observable networks* is a more realistic setting, in which only a certain fraction of nodes have both connections and node attributes, whereas the other nodes have either links or attributes unobservable. Consider the example in Figure 1. The network has 5 nodes with both link and attribute information and other nodes are partially observable. While the link information of node v_6 is missing, we could recommend potential friends from the candidate pool $\{v_2, v_3, v_5, v_9, v_{10}, v_{11}\}$ based on their attribute similarity. However, it would be more challenging to recommend from the candidates $\{v_1, v_4, v_7, v_8\}$ which only have link information. We refer to such problem as **Cross View Link Prediction (CVLP)**, in which we recommend nodes with only attributes to nodes with only links (or vice versa).

The CVLP task can be even more challenging in many real-world social/information networks, as node attributes are usually characterized by high dimensionality and contain certain amount of noisy/irrelevant attributes. For example, in Facebook network, one could extract millions of (sparse) features for user profiling, such as the groups a user has joined, the web pages he has liked, the content of posts, and the user's demographic features. Such high-dimensional features pose additional challenges to link prediction task. These features have different importances in predicting the links and some features might even have negative effect on the prediction. So it is critical to select only the relevant features for link prediction.

In this paper, we study the novel problem of CVLP, and propose an effective approach, Noise-resilient Representation Consensus Learning (NRCL), to address these challenges of cross view link prediction. Since nodes with only observable links and nodes with only observable attributes are not directly comparable in their original form, we propose to learn a common subspace in which nodes with incomplete information become comparable to each other. We utilize link-based representations and content-based representations of fully observable nodes to form a co-regularization consensus. Experimental results on real-world datasets demonstrate that NRCL outperforms baseline methods significantly. The contribution of the paper can be summarized as follows:

- To our best knowledge, we are the first to formulate and investigate the problem of cross-view link prediction on networks with partially observable links and node attributes.
- We propose to learn representation consensus so that nodes with either link information or node attributes could become comparable in the latent space. Two instantiations of the proposed framework, based on log loss and Huber loss, are developed and compared, with the latter being more robust to noisy link structure.
- Considering that many node attributes in real-world networks tend to be noisy/irrelevant, we perform joint feature selection in our framework to alleviate the issue of noisy attributes. To our knowledge, no prior work on node representation learning selects features jointly.
- We conduct experiments on four real-world networks and show the effectiveness of the proposed method on the task of cross-view link prediction.

The rest of the paper is organized as follows. In section 2, we briefly review related work on link prediction. In section 3, we provide some preliminary definitions for our framework. We present the robust framework for learning representation consensus in section 4 and 5. An alternating optimization framework for the proposed model is developed in section 6. Experimental results are shown in section 7.

2. RELATED WORK

The link prediction problem has been studied extensively in the data mining and machine learning community [1] [5] [29]. Various scoring methods have been proposed based on the topology of graphs: 1) Common Neighbor based methods: Adamic/Adar [1] assigns weight to each common neighbor based on the degree of the neighbors; 2) Path based methods such as Katz [9] and Local Path and Random Walk with Restart [14]. Katz [9] is a path based method which sums over all paths between two nodes.

Some link prediction methods [12] [2] formulate link prediction as a supervised task where the existence of link is used as supervision. For example, Lichtenwalter et al. studied how to ensemble different measures for link prediction [12]. Supervised Random Walk [2] is a random walk based approach to combine different similarity scores. It attempts to learn a weight for different features to make the transition probability between linked nodes larger than that of unlinked nodes.

Some work investigates the low rank approximation methods by generating a low rank matrix to approximate the adjacency matrix of network structure [24] [15]. Besides, various latent variable models [3] [29] [17] have been proposed to model the relationship between nodes. For example, WTFW [3], a topic model-based approach, can perform link prediction as well as providing explanation to support the prediction. Recently, embedding methods, such as DeepWalk [19], LINE [23] and node2vec [6], are developed to learn representations for network nodes based on the link structure. They employ similar objective function as the popular word embedding method Word2Vec [16] and the derived node embedding can be used for link prediction [6].

Recently, researchers study how to perform link prediction for the heterogeneous information network [22] [30] [28], where multiple types of nodes and links exist in the network.

However, existing methods usually assume the network structure is complete. No previous research studies cross-view link prediction on partially observable networks.

3. FORMULATIONS

In this section, we present a few preliminary definitions that will be used in the rest of this paper.

DEFINITION 1. **Information Network** *An information network $G = (V, E, X)$ consists of V, the set of nodes, $E \subseteq V \times V$, the set of links, and the feature matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ ($n = |V|$), where $\mathbf{x}_i \in R^D$ ($i = 1 \ldots n$) is the attribute vector of node v_i.*

Since many real-world social/information networks have partially observable links and node attributes, we study link prediction on such networks defined as follows.

DEFINITION 2. **Partially Observable Information Network** *In a partially observable information network $G = (V, E, X)$, each node can belong to one or two of the following (overlapping) sets: the set of nodes O^g with observable links and nodes O^a with observable attributes. We also use $O^s = O^g \cap O^a$ to denote the set of nodes with both observable links and attributes .*

Note that $V = O^g \cup O^a$ since we assume each node has at least one source of information. While it is possible that certain nodes have disclosed neither links nor attributes, we do not consider such nodes since no information can be used to suggest link to them in that case. In a partially observable information network $G = (V, E, X)$, many nodes have only one view of information, i.e., link or node attributes. We refer to nodes with only links ($O^g \backslash O^s$) as **link-only nodes** and nodes with only attributes ($O^a \backslash O^s$) as **attribute-only nodes**. In this paper, we study how to recommend link-only nodes to attribute-only nodes, or vice versa. We refer to such task as **Cross View Link Prediction (CVLP)**.

Let the number of nodes with links, nodes with attributes and nodes with both links and attributes be $n^g = |O^g|$, $n^a = |O^a|$ and $n^s = |O^s|$, respectively.

4. LINK-BASED REPRESENTATION LEARNING

We aim to learn representations for the network nodes by preserving structural information. For a node v_i, other nodes can be divided into to two classes, neighbors and non-neighbors. Hence, we can derive triplets (i, j, k) from the network structure, where v_i and v_j are neighbors while v_i and v_k are non-neighbors. We denote the set of all such triplets (i, j, k) as Ω.

Let us denote the representation learned from links as $\mathbf{U}^g \in \mathbb{R}^{n^g \times m}$, where m is the number of dimensions in the representation. The affinity s_{ij} between two nodes v_i and v_j can be calculated as the inner product of the representations $s_{ij} = \mathbf{U}^g_i (\mathbf{U}^g_j)^T$. To make the representation appropriate for link prediction, it is desirable to make the affinity between neighbors larger than the affinity between non-neighbors. So we aim to optimize the following objective.

$$\begin{aligned} &\min_{\mathbf{U}^g} \ ||\mathbf{U}^g||_F^2 \\ &\text{s.t. } s_{ij} \geq s_{ik}, \forall (i,j,k) \in \Omega \end{aligned} \tag{1}$$

This objective function minimizes the complexity of representation while keeping neighbors and non-neighbors separable. Since it might not be possible to satisfy all the hard constraint on all triplets (i, j, k), we minimize the number of mis-ordered ranking triplets. Let us denote $s_{ijk} = s_{ij} - s_{ik}$ and the objective function is the following.

$$\min_{\mathbf{U}^g} \sum_{(i,j,k)\in\Omega} \mathbf{I}(s_{ijk} < 0) + \lambda_g ||\mathbf{U}^g||_F^2 \tag{2}$$

where $\mathbf{I}(\cdot)$ is an indicator function which returns 1 if $(\cdot)$ is true and 0 otherwise. The 0/1 loss function is not smooth and is computationally intractable to optimize. So we replace it with a continuous convex surrogate loss $l(\cdot)$ in the objective function.

$$\min_{\mathbf{U}^g} \sum_{(i,j,k)\in\Omega} l(s_{ijk}) + \lambda_g ||\mathbf{U}^g||_F^2 \tag{3}$$

AUC (Area Under ROC Curve) is a widely used metric for evaluating binary prediction problem such as recommender system and link prediction [12]. It can be shown that optimizing the objective in Eq (2) is related to optimizing the AUC [20] [27]. Hence, learning the representation under such objective is a good choice for link prediction. There can be different options for the loss function $l(s_{ijk})$, such as log-loss, exponential loss and hinge loss. In the following subsection, we develop two instantiations of NRCL with different loss functions.

4.1 Probabilistic Representation Learning (P-RL)

From a generative point of view, one can assume all the triplets $(i, j, k) \in \Omega$ are generated from the node representation $\mathbf{U}^g$. More specifically, we model the probability of preserving ranking order $s_{ij} > s_{ik}$ using the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$.

$$P(s_{ij} > s_{ik} \mid \mathbf{U}^g) = \sigma(s_{ijk}) \tag{4}$$

The larger s_{ijk} is, the more likely ranking order $s_{ij} > s_{ik}$ is preserved. By assuming the ranking orders to be independent, the probability $P(> |\mathbf{U}^g)$ of all the ranking orders being preserved given $\mathbf{U}^g$ is the following.

$$\begin{aligned} P(> |\mathbf{U}^g) &= \prod_{(i,j,k)\in\Omega} P(s_{ij} > s_{ik} | \mathbf{U}^g) \\ &= \prod_{(i,j,k)\in\Omega} \sigma(s_{ijk}) \end{aligned} \tag{5}$$

So, the goal is to find the latent representation $\mathbf{U}^g$ for network nodes which maximizes $P(> |\mathbf{U}^g)$ (i.e., to make preserving the aggregated ranking orders have maximum probability). It can be performed by minimizing the following sum of negative log-likelihood:

$$\begin{aligned} \min_{\mathbf{U}^g} \ L^g &= -\log P(> |\mathbf{U}^g) + \lambda_g ||\mathbf{U}^g||_F^2 \\ &= -\sum_{(i,j,k)\in\Omega} \log P(s_{ij} > s_{ik} | \mathbf{U}^g) + \lambda_g ||\mathbf{U}^g||_F^2 \\ &= -\sum_{(i,j,k)\in\Omega} \log \sigma(s_{ijk}) + \lambda_g ||\mathbf{U}^g||_F^2 \end{aligned} \tag{6}$$

The connection between Eq (6) and Eq (3) is easy to see: log loss is used as the loss function $l(\cdot)$. Such a formulation provides a probabilistic interpretation for the ranking order preserving principle. Such a loss function is similar in spirit to the Bayesian Personalized Ranking [20], which attempts to predict the interaction between users and items.

4.2 Max Margin Representation Learning (MM-RL)

One can also employ a structural learning framework with max margin formulation as follows.

$$\begin{aligned} &\min_{\mathbf{U}^g} \sum_{(i,j,k)\in\Omega} \mu_{ijk} + \lambda_g ||\mathbf{U}^g||_F^2 \\ &\text{s.t. } s_{ijk} \geq 1 - \mu_{ijk}, \forall (i,j,k) \in \Omega \end{aligned} \tag{7}$$

where μ_{ijk} is a slack variable to impose soft margin. Such a formulation is similar to Structural SVM [8]. To make clear its connection to the Eq. (3) in the general framework, we can write it in the following form.

$$\min_{\mathbf{U}^g} \sum_{(i,j,k)\in\Omega} \max(0, 1 - s_{ijk}) + \lambda_g ||\mathbf{U}^g||_F \tag{8}$$

So, Eq. (8) is equivalent to using hinge loss as $l(\cdot)$ in Eq. (3).

The hinge loss is not differentiable at 0 and therefore poses difficulty for gradient-based optimization. We use a differentiable loss defined as follows.

$$l(x) = \begin{cases} 0 & \text{if } x \geq 2 \\ \frac{1}{4}(x-2)^2 & \text{if } 2 > x > 0 \\ 1 - x & \text{if } x \leq 0 \end{cases} \tag{9}$$

This loss function, which is also referred to as Huber loss, is a combination of L_1 loss (when $2 > x > 1$) and L_2 loss (when $x < 0$). In the link structure of many networks, there is often certain amount of noisy information. For example, it is not rare that a Facebook user may accept a connection invitation from someone he/she actually does not know (i.e., false positive), or two new users have not connected even if they know each other (i.e., false negative). Besides, the interaction between two proteins may have not been discovered due to the difficulty in the study of certain biological process (i.e., false negative). Such pairs might form noisy ranking triplets (i, j, k), which could potentially hamper the performance of link prediction models. Such noisy triplets might cause s_{ijk} to become negative. Rather than using L_2 loss on the whole range, Huber loss uses L_1 loss for $x < 0$ because L_1 loss penalizes the error less harshly than L_2 loss and hence more robust to noisy triplets.

Hence, the optimization problem becomes the following:

$$\min_{\mathbf{U}^g} L^g = \sum_{(i,j,k)\in\Omega} l(s_{ijk}) + \lambda_g ||\mathbf{U}^g||_F^2 \tag{10}$$

where $l(\cdot)$ is the Huber loss defined in Eq (9).

5. NOISE-RESILIENT REPRESENTATION CONSENSUS LEARNING (NRCL)

We have discussed how to learn ranking-based representation from network links with P-RL and MM-RL. In this section, we describe the framework of NRCL based on learning representation consensus. Linkage information alone might not be sufficient for learning node representation, since network links are often sparse and noisy. Also, the node features can be of high dimensionality and contain many irrelevant features. Since links and attributes provide complementary information on the network nodes, it is desirable to learn a consensus from the link-based representation and attribute-based representation. Also, the consensus learning enables link-only nodes and attribute-only nodes to be comparable in the latent space. Therefore, the similarity between the representations of two nodes can be used for cross view link prediction.

For the attribute-based representation, we learn a linear projection under the guidance of $\mathbf{U}^g$.

$$\min_{\mathbf{W}} \sum_{i\in O_s} ||\mathbf{U}_i^g - \mathbf{x}_i\mathbf{W}||_F^2 \tag{11}$$

If we represent all the $\mathbf{U}_i^g$ and $\mathbf{x}_i$ in $i \in O_s$ as $\mathbf{U}^s$ and $\mathbf{X}^s$, respectively, we can write the objective function in the following form.

$$\min_{\mathbf{W}} ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 \tag{12}$$

Different features usually have different importances for predicting the links. For example, in the Facebook social network, "went to the same college" could be a more informative feature than "live in the same country" for link prediction. The projection matrix $\mathbf{W}$ can encode such knowledge by optimizing the objective in Eq (12) and useful features tend to have large (absolute value of) weights in the matrix $\mathbf{W}$.

Besides, node features could contain many irrelevant ones which could even harm the representation learning. To address this challenge, we propose to perform joint feature selection when learning the projection. We use a feature selection indicator vector $\mathbf{s} \in \{0, 1\}^D$ where $s_p = 1$ indicates the p-th feature is selected and $s_p = 0$ indicates the feature is not selected.

$$\begin{aligned} \min_{\mathbf{W},\mathbf{s}} \ & ||\mathbf{U}^s - \mathbf{X}^s \operatorname{diag}(\mathbf{s})\mathbf{W}||_F^2 \\ \text{s.t. } & s_p \in \{0, 1\}, \forall p = 1, \ldots, D \\ & \sum_{p=1}^{D} s_p = d \end{aligned} \tag{13}$$

where $\operatorname{diag}(\mathbf{s})$ is the diagonal matrix with $\mathbf{s}$ as the diagonal elements. The constraint $\sum_{p=1}^{D} s_p = d$ means that we aim to select d ($d < D$) high quality features for the attribute-based representation. $\operatorname{diag}(\mathbf{s})\mathbf{W}$ is a matrix with d non-zero rows and hence it achieves feature selection. We combine $\mathbf{s}$ and $\mathbf{W}$ together, and employ $L_{2,0}$ norm to achieve the effect of feature selection:

$$\begin{aligned} \min_{\mathbf{W}} \ & ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 \\ \text{s.t. } & ||\mathbf{W}||_{2,0} \leq d \end{aligned} \tag{14}$$

The $L_{2,0}$ norm $||\mathbf{W}||_{2,0}$ is the number of rows in $\mathbf{W}$ with non-zero value. If $||\mathbf{W}_{i\cdot}||_F = 0$, i-th feature is not selected. The feasible region defined by $||\mathbf{W}||_{2,0} < d$ is not convex and we relax $||\mathbf{W}||_{2,0}$ to its convex hull:

$$\begin{aligned} \min_{\mathbf{W}} \ & ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 \\ \text{s.t. } & ||\mathbf{W}||_{2,1} \leq d \end{aligned} \tag{15}$$

where the $L_{2,1}$ norm $||\mathbf{W}||_{2,1} = \sum_{i=1}^{D} ||\mathbf{W}_{\mathbf{i}\cdot}||_F$ could also achieve row sparsity. We further write the constraint in the form of Lagrangian as follows:

$$\min_{\mathbf{W}} L^a = ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 + \lambda_a ||\mathbf{W}||_{2,1} \tag{16}$$

where λ_a is the regularization parameter on $L_{2,1}$ norm [18] [10].

We combine the link-based loss and attribute-based loss together and the objective function becomes the following:

$$\begin{aligned} \min_{\mathbf{U}^g,\mathbf{W}} L &= L^g + L^a \\ &= \sum_{(i,j,k)\in\Omega} l(s_{ijk}) + \lambda_g ||\mathbf{U}^g||_F^2 + \\ &\quad \alpha||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 + \lambda_a ||\mathbf{W}||_{2,1} \end{aligned} \tag{17}$$

where α is the parameter that controls the relative importance of consensus learning. We refer to the instantiations of NRCL with L_g in Eq (6) and Eq (10) as P-NRCL and MM-RNCL, respectively.

Figure 2 summarizes the NRCL framework: 1) Representation $\mathbf{U}^g$ learned on linkage information might not be sufficiently good, as network links could be sparse and noisy. The consensus constraint $||\mathbf{U}_i^g - \mathbf{U}_i^a||_F^2$ (where the attribute-based representation $\mathbf{U}_i^a = \mathbf{x}_i\mathbf{W}$) serves as additional regularization on $\mathbf{U}^g$, which enables link-based representation to incorporate information from node attributes. This can be especially useful when a node has very few

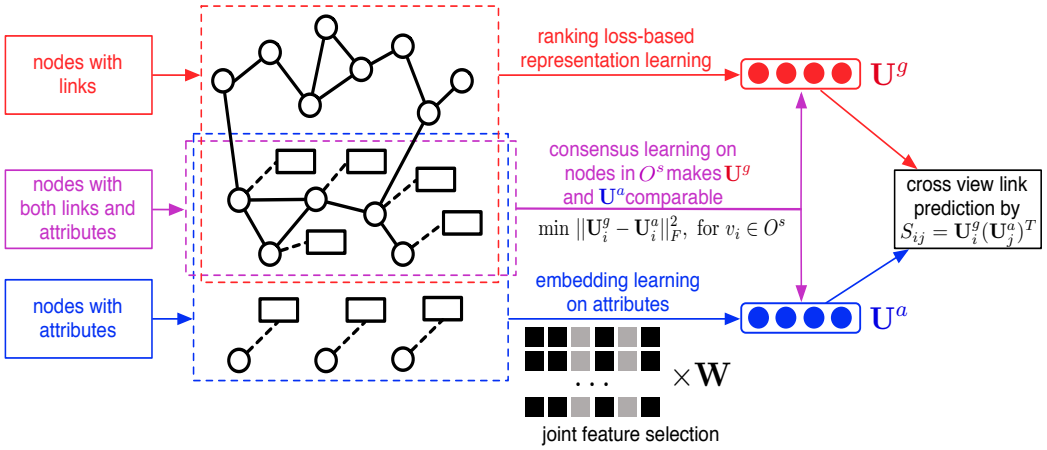


Figure 2: Representation consensus learning on partially observable networks

Algorithm 1 Alternating Optimization for NRCL

Initialize: $\mathbf{U}_i^g = rand(0,1)$, $\mathbf{W} = \mathbf{0}^{D\times m}$, $t = 1$.
while not converged **do**
  Fixing $\mathbf{W}$, find the optimal $\mathbf{U}^g$ by L-BFGS with Eq (25)
  Fixing $\mathbf{U}^g$, find the optimal $\mathbf{W}$ with Algorithm 2
  $t = t + 1$
end while
$\mathbf{U}_i^a = \mathbf{x}_i\mathbf{W}, \forall i \in O^a$

or no links. 2) On the other hand, node attributes are not equally important for link prediction. The consensus constraint $||\mathbf{U}_i^g - \mathbf{U}_i^a||_F^2$ can guide the learning of attribute-based representation by jointly selecting node attributes. By learning the consensus between $\mathbf{U}^g$ and $\mathbf{U}^a$, link information and attribute information could lend strength to each other for learning more noise-resilient representation. Also, the representations learned from network structure and node attributes become comparable in the latent space. To perform cross view link prediction, we can calculate the similarity $s_{ij} = \mathbf{U}_i^g(\mathbf{U}_j^a)^T$ in the latent space for link-only node v_i and attribute-only node v_j.

6. OPTIMIZATION

In this section, we discuss how to solve the optimization problem for P-NRCL and MM-NRCL.

6.1 Alternating Optimization

For both instantiations, we need to optimize over $\mathbf{U}^g$ and $\mathbf{W}$. We decompose it to two sub-problems and develop an alternating optimization approach to solve the problem.

6.1.1 *Fixing* $\mathbf{W}$*, update* $\mathbf{U}^g$

Now we derive the gradient for optimizing the objective function in Eq (6) and Eq (10). For P-NRCL, the gradient for one triplet is calculated as follows:

$$\frac{\partial l(s_{ijk})}{\partial \mathbf{U}_i^g} = \frac{e^{-s_{ijk}}}{1+e^{-s_{ijk}}} \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} \tag{18}$$

For Max Margin NRCL (MM-NRCL), the gradient is the following:

$$\frac{\partial l(s_{ijk})}{\partial \mathbf{U}_i^g} = \begin{cases} 0 & \text{if } s_{ijk} \geq 2 \\ \frac{1}{2}(s_{ijk}-2) \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} & \text{if } 2 > s_{ijk} > 0 \\ -\frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} & \text{if } s_{ijk} \leq 0 \end{cases} \tag{19}$$

The gradients on s_{ijk} w.r.t. $\mathbf{U}_i^g$, $\mathbf{U}_j^g$ and $\mathbf{U}_k^g$ are the following:

$$\frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} = \mathbf{U}_j^g - \mathbf{U}_k^g \tag{20}$$

$$\frac{\partial}{\partial \mathbf{U}_j^g} s_{ijk} = \mathbf{U}_i^g \tag{21}$$

$$\frac{\partial}{\partial \mathbf{U}_k^g} s_{ijk} = -\mathbf{U}_i^g \tag{22}$$

So, the gradient on L^g w.r.t $\mathbf{U}_i^g$ is as follows:

$$\begin{aligned} \frac{\partial L^g}{\partial \mathbf{U}_i^g} = &\sum_{(i,j,k)\in\Omega} \frac{e^{-s_{ijk}}}{1+e^{-s_{ijk}}} \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} + \\ &\sum_{(j,i,k)\in\Omega} \frac{e^{-s_{jik}}}{1+e^{-s_{jik}}} \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{jik} + \\ &\sum_{(j,k,i)\in\Omega} \frac{e^{-s_{jki}}}{1+e^{-s_{jki}}} \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{jki} \end{aligned} \tag{23}$$

We can also derive the following gradient on L^a w.r.t $\mathbf{U}_i^g$:

$$\frac{\partial L^a}{\partial \mathbf{U}_i^g} = 2\alpha(\mathbf{U}_i^g - \mathbf{U}_i^a) \tag{24}$$

where $\mathbf{U}_i^a = \mathbf{x}_i\mathbf{W}$. To sum up, the gradient of the objective function in Eq (17) w.r.t $\mathbf{U}_i^g$ is as follows:

$$\frac{\partial L}{\partial \mathbf{U}_i^g} = \begin{cases} \frac{\partial L^g}{\partial \mathbf{U}_i^g} & \text{for } \phi(i) \in O^g \backslash O^s \\ \frac{\partial L^g}{\partial \mathbf{U}_i^g} + \frac{\partial L^a}{\partial \mathbf{U}_i^g} & \text{for } \phi(i) \in O^s \end{cases} \tag{25}$$

We can use gradient-based method (e.g., steepest descent or L-BFGS) to solve this subproblem.

6.1.2 *Fixing* $\mathbf{U}^g$*, update* $\mathbf{W}$

With fixed $\mathbf{U}^g$, we find the optimal $\mathbf{W}$ for the following convex sub-problem.

$$\min_{\mathbf{W}} \mathcal{L} = ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 + \lambda'_a||\mathbf{W}||_{2,1} \tag{26}$$

where $\lambda'_a = \lambda_a/\alpha$. To solve this subspace learning with $L_{2,1}$ regularization, we develop Algorithm 2 inspired by the iterative approach used in [18].

By setting $\frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{W}} = 0$, we have the following:

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{W}} = (\mathbf{X}^s)^T(\mathbf{X}^s\mathbf{W} - \mathbf{U}^s) + \lambda'_a\mathbf{E}\mathbf{W} = 0 \Rightarrow \\ \mathbf{W} = ((\mathbf{X}^s)^T\mathbf{X}^s + \lambda'_a\mathbf{E})^{-1}(\mathbf{X}^s)^T\mathbf{U}^s\end{aligned} \tag{27}$$

where $\mathbf{E}$ is a diagonal matrix with diagonal elements $\mathbf{E}_{ii} = \frac{1}{2||\mathbf{W}_i||_F}$ and $\mathbf{W}_i$ is the i-th row of $\mathbf{W}$.

THEOREM 6.1. *For the optimization problem in Eq (26), Algorithm 2 would converge.*

Proof: It is easy to see that Eq (27) is a solution of the problem:

$$\min_{\mathbf{W}} ||\mathbf{X}^s\mathbf{W} - \mathbf{U}^s||_F^2 + \lambda'_a \operatorname{Tr}(\mathbf{W}^\mathbf{T}\mathbf{E}\mathbf{W}) \tag{28}$$

where $\operatorname{Tr}(\cdot)$ is the trace of matrix $(\cdot)$. So, from the t-th to $(t+1)$-th iteration,

$$\begin{aligned}&||\mathbf{X}^s\mathbf{W}^{t+1} - \mathbf{U}^s||_F^2 + \lambda'_a \operatorname{Tr}((\mathbf{W}^{t+1})^T\mathbf{E}^{t+1}\mathbf{W}^{t+1}) \\ &\leq ||\mathbf{X}^s\mathbf{W}^{t} - \mathbf{U}^s||_F^2 + \lambda'_a \operatorname{Tr}((\mathbf{W}^{t})^T\mathbf{E}^{t}\mathbf{W}^{t}) \Rightarrow \\ &||\mathbf{X}^s\mathbf{W}^{t+1} - \mathbf{U}^s||_F^2 + \lambda'_a \sum_i \frac{||\mathbf{W}_i^{t+1}||_F^2}{2||\mathbf{W}_i^t||_F} \\ &\leq ||\mathbf{X}^s\mathbf{W}^{t} - \mathbf{U}^s||_F^2 + \lambda'_a \sum_i \frac{||\mathbf{W}_i^{t}||_F^2}{2||\mathbf{W}_i^t||_F}\end{aligned} \tag{29}$$

Equivalently,

$$\begin{aligned}&||\mathbf{X}^s\mathbf{W}^{t+1} - \mathbf{U}^s||_F^2 + \lambda'_a||\mathbf{W}^{t+1}||_{2,1} - \\ &\lambda'_a(||\mathbf{W}^{t+1}||_{2,1} - \sum_i \frac{||\mathbf{W}_i^{t+1}||_F^2}{2||\mathbf{W}_i^t||_F}) \\ &\leq ||\mathbf{X}^s\mathbf{W}^{t} - \mathbf{U}^s||_F^2 + \lambda'_a||\mathbf{W}^{t}||_{2,1} - \\ &\lambda'_a(||\mathbf{W}^{t}||_{2,1} - \sum_i \frac{||\mathbf{W}_i^{t}||_F^2}{2||\mathbf{W}_i^t||_F})\end{aligned} \tag{30}$$

Note that $||\mathbf{W}^{t+1}||_{2,1} - \sum_i \frac{||\mathbf{W}_i^{t+1}||_F^2}{2||\mathbf{W}_i^t||_F} \leq ||\mathbf{W}^{t}||_{2,1} - \sum_i \frac{||\mathbf{W}_i^{t}||_F^2}{2||\mathbf{W}_i^t||_F}$ (because $\sqrt{a} - \frac{a}{2\sqrt{b}} \leq \sqrt{b} - \frac{b}{2\sqrt{b}}, a, b > 0$). So,

$$\begin{aligned}\mathcal{L}(\mathbf{W}^{t+1}) &= ||\mathbf{X}^s\mathbf{W}^{t+1} - \mathbf{U}^s||_F^2 + \lambda'_a||\mathbf{W}^{t+1}||_{2,1} \\ &\leq ||\mathbf{X}^s\mathbf{W}^{t} - \mathbf{U}^s||_F^2 + \lambda'_a||\mathbf{W}||_{2,1} = \mathcal{L}(\mathbf{W}^t)\end{aligned} \tag{31}$$

The objective function $\mathcal{L}(\mathbf{W})$ decreases in each iteration and it is lower bounded, so the convergence of Algorithm 2 is guaranteed. In our experiments, we observe it converges usually in less than 10 iterations.

Algorithm 1 summarizes the optimization methods for NRCL. The following theorem shows the convergence of this algorithm.

THEOREM 6.2. *The alternating optimization framework in Algorithm 1 would converge.*

Proof: The objective function for each subproblem decreases in each iteration. The objective function in Eq (17) is hence guaranteed to decrease and it is lower-bounded. So the alternating optimization algorithm 1 would converge.

Algorithm 2 Algorithm for Learning Projection with $L_{2,1}$ norm

1: **Input**: $\mathbf{X^s} \in \mathcal{R}^{n_s \times D}$, projection target $\mathbf{U^s} \in \mathcal{R}^{n_s \times m}$, λ'_a
2: **Initialize**: $\mathbf{E} = \mathbf{I}_D$
3: **while** not converged **do**
4: $\quad \mathbf{W} = ((\mathbf{X}^s)^T\mathbf{X} + \lambda'_a\mathbf{E})^{-1}(\mathbf{X}^s)^T\mathbf{U}^s$
5: $\quad \mathbf{E} = \begin{bmatrix} \frac{1}{2||\mathbf{W}_1||_F} & & \\ & \cdots & \\ & & \frac{1}{2||\mathbf{W}_D||_F} \end{bmatrix}$
6: **end while**
7: **Output**: $\mathbf{W} \in \mathcal{R}^{D \times m}$

Table 1: Statistics of datasets

Statistics	Blogcatalog	Facebook	Wiki	Pubmed
# of instances	3192	1045	3363	19717
Avg. degree	8.87	51.19	19.76	4.50
# of attributes	3221	576	4973	500

6.2 Sampling ranking triplets

One can derive $O(|E||V|)$ triplets from the network structure. Such large amount of triplets is computationally expensive to optimize on. Rather than using all the potential triplets, we only sample a portion of them as follows: for each link (v_i, v_j) in the network, we randomly sample n_k ($n_k \ll |V|$) negative pairs to form triplets with (v_i, v_j). Hence, a total of $|E|n_k$ triplets (i.e, $|\Omega| = |E|n_k$) is used in the optimization. In our preliminary experiments, we found $n_k = 2$ or $n_k = 3$ is usually sufficient to achieve decent performance, so we use $n_k = 2$ in our experiments.

7. EXPERIMENTS

In this section, we perform cross-view link prediction on four real-world networks with partially observable links and node attributes.

7.1 Datasets

We use four publicly available social/information network datasets in our experiments:

- Facebook Dataset[3]: The whole dataset consists of ten ego-networks of facebook users. We use the network with largest number of nodes, which has 1045 users, 576 user profile features (e.g., education, work and location) and 53498 links.
- BlogCatalog Dataset[4]: We extract users who have blog posts in the category of {Music, Finance, Health, Computers, Entertainment}. The friendship connection between blog users establishes the network links and the word occurrence in the blogs is used as user features.
- Wikipedia Dataset[5] [21]: Wikipedia articles from 19 categories and the hyperlinks establish the network structure. The original hyperlinks are directed and we symmetrize the network to make it undirected.

[3] https://snap.stanford.edu/data/egonets-Facebook.html
[4] http://dmml.asu.edu/users/xufei/datasets.html
[5] http://linqs.cs.umd.edu/projects//projects/lbc/index.html

- PubMed Dataset[5] [21]: It consists of 19717 scientific publications about diabetes from PubMed database, which are classified into one of three classes. The word occurrence in the paper is used as the node features.

The statistics of these datasets are shown in Table 1.

7.2 Experimental Setting

7.2.1 Baselines

Existing methods usually assume the completeness of links and are not directly applicable for our problem setting. We create content links for each node in O^a (that has attributes) by connecting them with k other nodes with largest similarity w.r.t attributes, where k is the average degree of the network. Then we construct a combined network by connecting two nodes when they have either a structural link or content link between them. We use the following methods on this combined network:

- Probabilistic Representation Learning (P-RL): P-RL learns the repsentation of nodes by optimizing the objective function L_g in Eq. (6), which is similar to the triple loss based link prediction [15] [20].
- Max Margin Representation Learning(MM-RL): MM-RL learns the representation by optimizing the objective function L_g in Eq. (10).
- LINE: An efficient embedding learning approach for network nodes [23] and the similarity between node embeddings can be used for link prediction.
- DeepWalk: It learns node representations that encode structural information by using truncated random walk as input [19]. Recent work shows that it has state-of-the-art performance for link prediction [6].

7.2.2 Evaluation Metrics

We use the widely adopted metrics Precision, Recall to evaluate the performance of different link prediction approaches.

- $Precision@N = \frac{|C_{R_N} \cap C_{adopted}|}{N}$
- $Recall@N = \frac{|C_{R_N} \cap C_{adopted}|}{|C_{adopted}|}$

where C_{R_N} is the set of top N nodes in the recommendation and $C_{adopted}$ is the set of links that actually exist in the network. The precision and recall averaged over all the nodes are reported to reflect performance of each link prediction approaches.

7.2.3 Generating Partially Observable Networks

To create partially observable networks, we randomly select a few nodes (the number of these nodes is denoted as m_1) and remove their links. After removing these links, we denote the number of nodes without any links as m_2. Typically m_2 is larger than m_1 since removing the links for the m_1 nodes may also make some other nodes become isolated. Then we pick another m_2 nodes randomly which have links and remove their attributes. Hence, only $|O^s| = n - 2m_2$ nodes have both links and attributes. For recommending attribute-only nodes to link-only nodes (or link-only to attribute-only), 20% of the link-only (or attribute-only) nodes is used for validation and the rest is used for testing.

We set the dimension sizes (i.e., m) of P-NRCL, MM-NRCL, P-RL, MM-RL to 50 and that of LINE and DeepWalk to their default setting. For the regularization parameters in P-NRCL, MM-NRCL, P-RL and MM-RL, we perform grid search on the validation set in the following ranges: $\alpha = \{0.01, 0.1, 1, 10\}$, $\lambda_a = \{10, 20, 30\} \times \alpha$, $\lambda_g = \{5, 10, 15, 20\}$.

7.3 Comparison on Cross View Link Prediction

We report the link prediction performance with different percentages ($|O_s|/n$) of fully observable nodes in Table 2 by setting $m_1 = \{0.2, 0.3\} \times n$. On most of the datasets, NRCL methods (especially MM-NRCL) outperform the baseline methods significantly. For example, on Wiki dataset, P-NRCL and MM-NRCL improve over the best baseline method MM-RL by 29.1% and 44.8%, respectively, in terms of precision@5. When the fully observable rate goes to as low as $20\% \sim 40\%$, MM-NRCL still performs very well for cross view link prediction. Though MM-RL and P-RL employ the same objective function L_g on the link-based representation learning as MM-NRCL and P-NRCL, the content links created from potentially noisy feature space make them unable to learn high quality representation. This indicates the importance of selecting the most informative features for representation learning on partially observable networks, in order to achieve decent link prediction performance. In comparison, the representation learned by MM-NRCL and P-NRCL could be more resilient to irrelevant features, as NRCL performs joint feature selection and only use the high-quality features to learn the representation.

When comparing P-NRCL with MM-NRCL, we observe that MM-NRCL performs better than P-NRCL in most cases. Similarly, MM-RL often outperforms P-RL as well. This suggests that Huber loss, which is more robust to noisy links, tends to be a better choice for learning node representations than log loss.

7.4 Case Study on Joint Feature Selection

Since we perform joint feature selection in our NRCL framework, the utility of feature i ($i = 1, 2, \ldots, D$) can be ranked by their coefficients $\sum_{j=1}^{m} W_{ij}^2$. For useless features, $\sum_{j=1}^{m} W_{ij}^2$ tends to shrink towards zero under the effect of $L_{2,1}$ norm, while important features tend to have large values of $\sum_{j=1}^{m} W_{ij}^2$. As a case study, we show the feature importance for Facebook dataset in Table 3. The specific value and meaning of features are anonymized for privacy concern. Features in the same category (e.g., education) can be encoded into multiple binary features and they may have different importance for representation learning. For instance, *education features* 538 and 237 are highly important while *education feature* 459 is considered useless. By examining the features with large scores ($\sum_{j=1}^{m} W_{ij}^2$), one could have a deeper understanding about the roles of different features in the formation of network links.

7.5 Sensitivity Analysis

In this subsection, we study the effect of dimension size m and consensus regularization parameter α only for MM-NRCL, since previous results show that MM-NRCL is more promising than P-NRCL. The precision results w.r.t different parameter values on Facebook and Wiki datasets are shown in Figure 3.

Effect of latent dimension size For m, we can observe that MM-NRCL is not very sensitive to the parameter value and performs consistently well when it is not too small (e.g., $m \leq 20$).

Effect of regularization controlling consensus strength For the co-regularization parameter α, MM-NRCL can achieve good performance as long as α is not too large (e.g., $\alpha \geq 10$).

Table 2: Link prediction with different observable rates

Dataset	$\|O_s\|/n$	Metric	Recommend AO to LO						Recommend LO to AO					
Facebook	0.5770		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	39.47	**44.87**	34.34	41.58	27.24	8.68	21.53	**23.57**	16.69	17.71	11.85	8.79
		Recall@5 (%)	16.33	**20.09**	13.62	19.94	7.92	4.92	10.62	**12.41**	7.82	8.53	5.25	4.15
	0.3703		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	38.45	**39.48**	32.96	32.10	30.47	11.33	20.24	**22.51**	16.76	17.33	14.09	8.91
		Recall@5 (%)	10.26	**10.86**	8.03	9.66	6.65	3.31	6.63	**8.35**	4.59	5.52	3.84	4.13
BlogCatalog	0.5081		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	**14.65**	13.74	0.90	2.19	7.23	0.65	**1.18**	1.12	0.11	0.28	0.06	0.45
		Recall@5 (%)	**33.27**	30.82	2.12	4.68	15.08	1.59	**2.86**	1.93	0.23	0.66	0.14	1.01
	0.2701		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	13.43	**13.51**	0.41	1.45	8.24	0.74	**0.66**	0.44	0.09	0.13	0.06	0.63
		Recall@5 (%)	24.61	**24.83**	0.79	1.97	15.58	1.38	0.58	**0.59**	0.10	0.10	0.01	1.11
Wiki	0.5332		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	21.77	**24.42**	13.05	16.86	10.18	3.58	11.49	**14.12**	6.59	10.33	2.81	2.94
		Recall@5 (%)	25.91	**29.28**	11.50	16.52	11.31	4.49	14.78	**18.46**	7.75	12.51	3.11	2.58
	0.3417		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	21.01	**24.31**	13.23	17.83	9.90	3.59	11.05	**13.61**	3.96	7.87	1.09	3.51
		Recall@5 (%)	20.42	**25.48**	10.48	16.13	9.98	3.02	11.20	**14.18**	3.42	6.72	0.89	3.22
PubMed	0.4521		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	3.18	**4.90**	0.89	1.50	0.02	0.98	0.69	1.03	0.17	0.28	0.05	**1.44**
		Recall@5 (%)	7.92	**12.60**	1.61	2.93	0.03	2.31	1.57	2.63	0.22	0.57	0.18	**3.43**
	0.1847		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	2.57	**4.34**	0.41	1.08	0.03	0.82	0.44	0.69	0.10	0.19	0.02	**1.05**
		Recall@5 (%)	4.77	**9.66**	0.55	1.30	0.06	1.94	0.85	1.38	0.19	0.28	0.03	**2.38**

Table 3: Feature importance on Facebook dataset

Feature Name	Feature Score
Top ranked features	
education;school;id;anonymized feature 538	1.1095
education;school;id;anonymized feature 237	0.4649
work;employer;id;anonymized feature 151	0.4579
education;school;id;anonymized feature 52	0.3724
education;concentration;id;anonymized feature 14	0.3499
Examples of unselected features	
last_name;anonymized feature 592	0
work;employer;id;anonymized feature 648	0
work;position;id;anonymized feature 697	0
work;end_date;anonymized feature 674	0
education;school;id;anonymized feature 459	0

8. CONCLUSION

In many real-world networks, the links and node attributes are often partially observable. In this paper, we study how to recommend link-only nodes to attribute-only nodes (or vice versa). To perform such cross-view link prediction, we propose to learn a representation consensus between links and attributes. Two instantiations that employ different ranking-based loss are presented for the representation learning. Considering high-dimensional node attributes are potentially noisy, we perform joint feature selection in the representation learning process. The link-based representation and the attribute-based representation could lend strength to each other and make the representation more resilient to link and attribute noise. Experimental results shows that the proposed P-NRCL and MM-NRCL are able to learn high-quality representations, which can effectively perform cross-view link prediction.

9. REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.

[3] N. Barbieri, F. Bonchi, and G. 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, 2014.

[4] F. Bonchi, C. Castillo, A. Gionis, and A. Jaimes. Social network analysis and mining for business applications. *ACM Trans. Intell. Syst. Technol.*, 2(3):22:1–22:37, 2011.

[5] Z. Chen, M. Chen, K. Q. Weinberger, and W. Zhang. Marginalized denoising for link prediction and multi-label learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1707–1713, 2015.

[6] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.

[7] M. A. Hasan and M. J. Zaki. A survey of link prediction in social networks. In *Social Network Data Analytics*, pages 243–275. 2011.

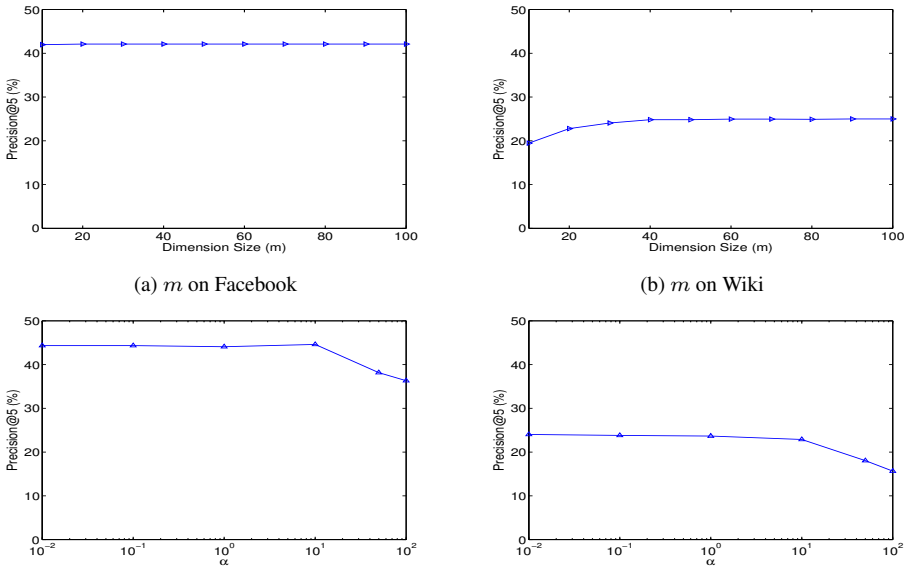
[8] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

[9] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, VOL. 18, NO. 1:39– 43, 1953.

[10] J. Li, X. Hu, L. Wu, and H. Liu. Robust unsupervised feature selection on networked data. In *Proceedings of the 2016 SIAM International Conference on Data Mining, Miami, Florida, USA, May 5-7, 2016*, pages 387–395, 2016.

[11] D. Liben-Nowell and J. M. Kleinberg. The link prediction problem for social networks. In *CIKM*, pages 556–559, 2003.

[12] R. Lichtenwalter, J. T. Lussier, and N. 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, 2010.



(a) m on Facebook

(b) m on Wiki

(c) α on Facebook

(d) α on Wiki

Figure 3: Parameter sensitivity for MM-NRCL

[13] W. Lin, X. Kong, P. S. Yu, Q. Wu, Y. Jia, and C. Li. Community detection in incomplete information networks. In *Proceedings of the 21st International Conference on World Wide Web*, pages 341–350. ACM, 2012.

[14] L. Lu and T. Zhou. Link prediction in complex networks: A survey. *Physica A*, 390(6):1150–1170, 2011.

[15] A. K. Menon and C. Elkan. Link prediction via matrix factorization. In *Proceedings of the ECML/PKDD 2011*, 2011.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[17] K. T. Miller, T. L. Griffiths, and M. I. Jordan. Nonparametric latent feature models for link prediction. In *NIPS*, pages 1276–1284, 2009.

[18] F. Nie, H. Huang, X. Cai, and C. H. Q. Ding. Efficient and robust feature selection via joint l2, 1-norms minimization. In *NIPS*, pages 1813–1821, 2010.

[19] B. Perozzi, R. Al-Rfou', and S. 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.

[20] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[21] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[22] Y. Sun, J. Han, C. C. Aggarwal, and N. V. Chawla. When will it happen?: relationship prediction in heterogeneous information networks. In *WSDM*, pages 663–672. ACM, 2012.

[23] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, 2015.

[24] F. Wang, T. Li, X. Wang, S. Zhu, and C. H. Q. Ding. Community discovery using nonnegative matrix factorization. *Data Min. Knowl. Discov.*, 22:493–521, 2011.

[25] X. Wei, B. Cao, W. Shao, C.-T. Lu, and P. S. Yu. Community detection with partially observable links and node attributes. In *IEEE International Conference on Big Data*, 2016.

[26] X. Wei, B. Cao, and P. S. Yu. Unsupervised feature selection on networks: A generative view. In *AAAI*, 2016.

[27] X. Wei, S. Xie, and P. S. Yu. Efficient partial order preserving unsupervised feature selection on networks. In *SDM*, pages 82–90. SIAM, 2015.

[28] L. Xu, X. Wei, J. Cao, and P. S. Yu. Embedding of embedding (eoe) : Embedding for coupled heterogeneous networks. In *WSDM*, 2017.

[29] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu. Stochastic relational models for discriminative link prediction. In *NIPS*, pages 1553–1560, 2006.

[30] J. Zhang, P. S. Yu, and Z.-H. Zhou. Meta-path based multi-network collective link prediction. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1286–1295, 2014.