On Exploring Semantic Meanings of Links for Embedding Social Networks

Linchuan Xu
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
cslcxu@comp.polyu.edu.hk

Xiaokai Wei*
Facebook Inc.
1 Hacker Way, Menlo Park, CA, USA
weixiaokai@gmail.com

Jiannong Cao
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
csjcao@comp.polyu.edu.hk

Philip S. Yu
University of Illinois at Chicago & Institute for Data Science, Tsinghua University
Chicago, Illinois, USA
psyu@uic.edu

ABSTRACT

There are increasing interests in learning low-dimensional and dense node representations from the network structure which is usually high-dimensional and sparse. However, most existing methods fail to consider semantic meanings of links. Different links may have different semantic meanings because the similarities between two nodes can be different, e.g., two nodes share common neighbors and two nodes share similar interests which are demonstrated in node-generated content. In this paper, the former type of links are referred to as structure-close links while the latter type are referred to as content-close links. These two types of links naturally indicate there are two types of characteristics that nodes expose in a social network. Hence, we propose to learn two representations for each node, and render each representation responsible for encoding the corresponding type of node characteristics, which is achieved by jointly embedding the network structure and inferring the type of each link. In the experiments, the proposed method is demonstrated to be more effective than five recent methods on four social networks through applications including visualization, link prediction and multi-label classification.

CCS CONCEPTS

• **Information systems → Data mining**; **Social networks**;

KEYWORDS

Network embedding; social networks; data mining

ACM Reference Format:
Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S. Yu. 2018. On Exploring Semantic Meanings of Links for Embedding Social Networks. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186114

1 INTRODUCTION

Social networks can meet and even create social needs for human beings as it is very convenient for people to reach out and get connected to others. We can analyze data regarding social network involvement to help improve networks and user experiences. This may explain why there are increasing interests from both industry and academia in personalizing services to users' interests, behaviors and attributes [14] [5], detecting communities [10, 33], recommending products [23, 43] or friends [12, 34–36]. However, the network representation, e.g., adjacency matrix, is usually high-dimensional and spare. High-dimensionality makes it inefficient to train data mining models while sparsity makes it not easy to generalize trained models for future usage.

Recently, network embedding [11, 21, 25, 37–39] has been utilized to learn low-dimensional and dense node representations basically by embedding the network structure in a certain Euclidean space. Although being demonstrated effective to preserve the network structure, almost all existing methods fail to consider semantic meanings of links in a social network where different links may have different semantic meanings. Semantically different links result from different reasons for which links are established and the reasons can be of various types in practice, e.g., two persons share a common university friend and two persons are fans of Star Wars. Hence, semantically different links may indicate nodes expose different types of information in a social network, e.g., the former example mentioned above may indicate the two persons expose information about their education while the latter example may indicate they expose information about their interests. As a result, it is more reasonable to embed the network structure with the semantic meanings of each link considered.

In this paper, we thus take the first step towards exploring semantic meanings of social links in the context of network embedding. Specifically, we propose to categorize social links according to their semantic meanings and learn node representations with semantic concepts. Inspired by homophily principle [18] which suggests links exist between nodes with similarities, we categorize links according to the type of similarities two nodes share. As the first work, this paper only considers two types of coarse-grained similarities, i.e., common neighbors and similar interests demonstrated in node-generated content, which represent two most frequently

[*] The work was done when the author was a Ph.D. student at University of Illinois at Chicago.

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186114

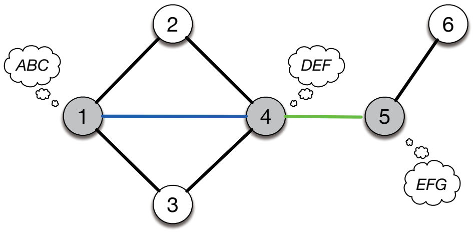

Figure 1: Structure-close link between node 1 and node 4 while content-close link between node 4 and node 5. Capital letters in cloud-like box denote node-generated content.

used types of similarity measurement, i.e., topology-based similarities and node content-based similarities, for two nodes to have a link [1, 15]. If two nodes share common neighbors, the link can be referred to as a *structure-close* link while if they share interests, the link can be referred to as a *content-close* link as illustrated in Fig. 1. It is worthy of noting that a particular link can be both structure-close and content-close since the two nodes can have common neighbors and share similar content at the same time.

We further generalize the concept of structure-close links between two nodes that are close in the network structure, e.g., two nodes can be close when they are connected by third-order links or even fourth-order links (second-order links exist between two nodes sharing common neighbors). But we do not set a threshold for the order of links to define the structure-close links because the threshold is not that important in this paper.

The existence of two types of semantic meanings of links suggests that it may not be proper to learn a single representation for each node because they may indicate the existence of two types of node characteristics, e.g., education and interests mentioned above. Education represents the type of characteristics that are intrinsic to nodes while interests represent the type nodes acquire from the particular social environment. The problem with a single representation is determined by the nature of network embedding, i.e., nodes connected by links should be close in the embedding space of interest. Accordingly, two nodes with different types of characteristics may be wrongly presented to be close.

For example, in Fig. 1, since node 5 is connected to node 4 and node 4 is connected to node 1, the representation of node 5 may also be close to node 1 in the embedding space, which may suggest node 5 and node 1 have a potential link. However, if the structure-close link between node 1 and node 4 is established because they share many university alumni while node 5 is not one of the alumni, node 1 and node 5 are not much likely to have a link because they also do not share interests. In this example, we make several assumptions about the links and node-generated content. One may question the assumption that node 1 and node 4 are connected by a structure-close link but do not share content. To validate this assumption, we present density of cosine similarity between tweets of two users connected by a structure-close link whose nodes share neighbors in a Twitter network described in Section 6 and corresponding cumulative density in Fig. 2. We can see there is a large portion of links whose nodes share little or even no common content.

In this paper, we thus propose to learn two representations for each node and render each representation responsible for encoding a particular type of node characteristics. Since content-close links mainly result from sharing similar interests, we name node characteristics exposed in these links as interests, and the corresponding representation as interest representation. For structure-close links resulting from sharing neighbors, we may think each node has an idea of whom the other node is by referring to its neighbors. Hence, we name the other type of node characteristics as identity and the corresponding representation as identity representation. The proposed method is thus referred to as interest and identity representation learning (IIRL).

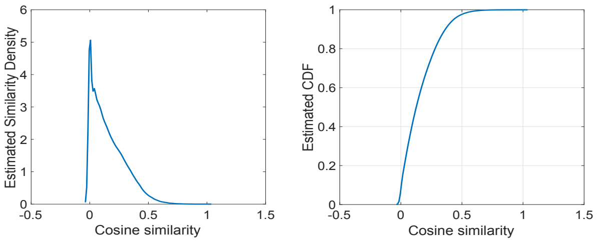

Figure 2: Similarity density plot and cumulative density plot

With the two representations, the problem faced by learning a single representation mentioned above can be solved as follows: node 1 and 4 are presented to be close in the embedding space corresponding to identity representations while node 4 and 5 are close in the space for interest representations. As a result, node 1 may not be close to node 5 because they are far away from each other in both of the spaces. Moreover, with these two salient semantic meanings, the two representations can make natural data mining tasks, e.g., link prediction and classification, more interpretable than single representations without semantic meanings [8].

The idea of learning two representations imposes two major challenges. Firstly, unlike learning a single representation that preserves both interest-related and identity-related characteristics, IIRL has to make sure that these two types of characteristics are preserved where they should be. But each link can be both structure-close and content-close. Hence, the challenge is how to infer the type of node characteristics exposed in each link so that the network structure and semantic meanings of links are well preserved in node representations. Secondly, besides the network structure, we also need to encode node-generated content into interest representations.

To address the first challenge, we infer the responsibility weight for each type of node characteristics given a particular link. The responsibility weights are estimated because characteristics in a particular link can be an arbitrary mixture of interest-related characteristics and identity-related ones. For the second challenge, we use node content as guidance while learning the interest representation as the content has the ground truth about node interests.

The contributions of the paper are summarized as follows:

1. To our best knowledge, this is the first work to explore semantic meanings of social links in network embedding.
2. We propose IIRL to learn two representations for each node by embedding the network structure and node-generated content where the network structure is gracefully embedded via joint link type inference.
3. We shed light on understanding social behaviors of individual users and an entire social network as a by-product of embedding social networks.

4. Via comprehensive evaluation on four social networks, we show data mining models built on the two representations are more accurate as well as more interpretable .

2 RELATED WORK

The development of recent network embedding starts with DeepWalk [21], which employs Skip-gram [19], a language model, to present pairs of nodes reached in truncated random walks to be close in the embedding space. With the success of DeepWalk, there emerge a series of Skip-gram based methods, such as TADW [40] to embed both network structure and node attributes, and node2vec [11] to explore diverse neighborhoods of each node.

There are also many other kinds of methods. LINE [25] is proposed to embed large-scale networks by directly presenting pairs of nodes with first-order or second order connections to be close. GraRep [6] models first-order up to a pre-defined k-order proximities into transition matrices, and then factorizes each transition matrix using SVD. A recent method [41] concludes that modeling high-order proximities can improve the quality of node representations, and then makes improvement to aforementioned models. Besides simply preserving the network structure, some methods also preserve network properties, such as HOPE [20] preserving asymmetric transitivities and M-NMF [30] preserving communities. Some methods [7] [24] [37] [9] even embed heterogeneous networks including more than one types of nodes and edges. Deep learning models have also been applied for network embedding [29]. Most of the methods above are unsupervised learning methods. Semi-supervised methods [28] [42] [13] have also been studied since representations with label information can perform better in subsequent classification tasks.

However, none of the methods mentioned above consider semantic meanings of links in a social network.

3 PRELIMINARIES

DEFINITION 1. *A **social network with node-generated content** is denoted as $G(N, E, A)$, where N is a set of nodes, E is a set of weighted or unweighted, directed or undirected edges. The type of node characteristics, i.e., interest-related characteristics and identity-related characteristics, in each edge is unknown. $A \in \mathbb{R}^{M \times L}$ is a term frequency matrix of attributes extracted from the content where M is the number of nodes and L is the number of attributes. It is assumed that each node content is non-empty.*

As a network embedding model, IIRL learns node representations through embedding first-order linkage information and non-linkage information of $G(N, E, A)$ in an Euclidean space. More specifically, each pair of nodes connected by an edge is presented to be close while each pair not connected is presented apart. Since there are two representations for each node, i.e., identity representation and interest representation, there are two corresponding spaces. The closeness of two nodes in the identity space is quantified as follows:

DEFINITION 2. *The **closeness** of two nodes in the identity Euclidean space is quantified as the probability of a structure-close edge between them, where the probability is defined as follows:*

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

where e_{ij_u} is a structure-close edge, $\boldsymbol{u}_i \in \mathbb{R}^{D_u}$ and $\boldsymbol{u}_j \in \mathbb{R}^{D_u}$ are column vectors of identity representations for nodes i and j, respectively, and D_u is the dimension of the identity Euclidean space. In the rest of the paper, the closeness is referred to as identity closeness.

Eq. (1) is symmetric in terms of node representations. For directed edges, the asymmetric property is not considered as LINE [25] and EOE [37] do. We may consider it in future work. Interest closeness is similarly defined. The closenesses can be quantified in this way because larger probabilities indicate larger inner products of two vectors, and the inner product is a measurement of closeness of two points in Euclidean space. Moreover, the closeness quantified in this way is to facilitate the formulation of an optimization objective presented in the following section.

Although the definition of closeness is similar to that of LINE(1st) [25] and EOE [37], IIRL further assigns a responsibility weight to the type of node characteristics in each edge, which is reflected on the overall closeness. The overall closeness of two nodes is summarized over weighted identity closeness and weighted interest closeness as follows:

$$p(e_{ij}) = \pi_{ij_u} p(e_{ij_u}) + \pi_{ij_v} p(e_{ij_v}), \tag{2}$$

where $p(e_{ij_v})$ quantifies interest closeness, e_{ij_v} is a content-close edge, and π_{ij_u} and π_{ij_v} are defined as follows:

DEFINITION 3. ***Identity proportion** of a linkage relationship denoted as π_{ij_u} quantifies the responsibility weight of identity-related characteristics in the link between node i and node j. Similarly, **interest proportion** of a linkage relationship denoted as π_{ij_v} quantifies the responsibility weight of interest-related characteristics. As there are only two types of node characteristics as indicated in the introduction, $\pi_{ij_u} + \pi_{ij_v} = 1$.*

According to Eq. (2), to correctly estimate the responsibility weights of the linkage relationship between two nodes, interest proportion is expected to be larger than identity proportion when the probability estimated by interest representations is larger than that estimated by identity representations, and vice versa. Otherwise, the weighted sum of the probability is less optimal. This expectation is consistent with our intuition that the stronger one out of two characteristics is more likely to dominate the relationship of two persons, e.g., the same attitude to Golden State Warriors or Cleveland Cavaliers may be more likely than the same born state, even the same university of two US citizens to dominate their relationships. And this intuition sheds light on how IIRL infers the link type.

4 IIRL MODEL

To obtain optimal node representations and link types for embedding the network structure, the optimization objective can be formulated according to maximum likelihood estimation since closeness is quantified by a concept of probability. Hence, the network structure embedding is formally formulated as the following problem:

$$\begin{gathered}\max_{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\pi}} \prod_{e_{ij} \in E} p(e_{ij}) \prod_{e_{hk} \notin E} (1 - p(e_{hk})) \\ s.t.\ \pi_{ij_u} + \pi_{ij_v} = 1 \,\&\, \pi_{hk_u} + \pi_{hk_v} = 1,\end{gathered} \tag{3}$$

where $\boldsymbol{U} \in \mathbb{R}^{M \times D_u}$ and $\boldsymbol{V} \in \mathbb{R}^{M \times D_v}$ is the matrix of identity representations and interest representations, respectively, and $\boldsymbol{\pi}$

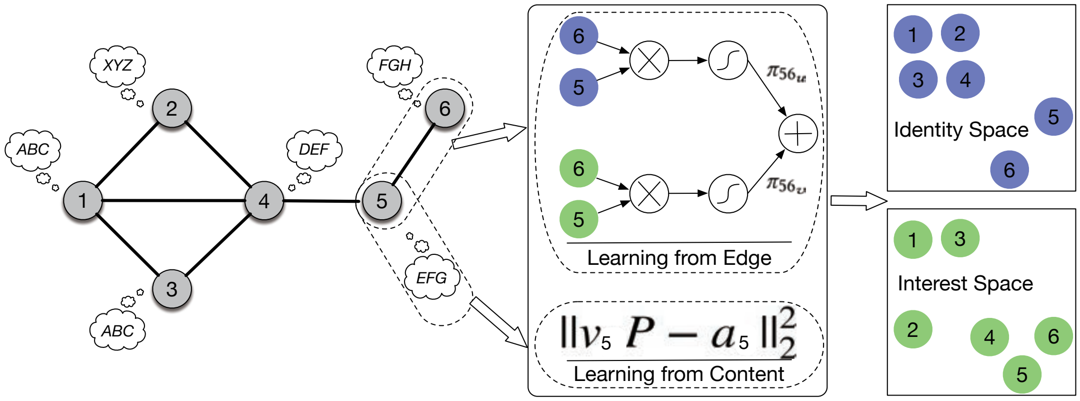

Figure 3: An illustration of embedding a toy network by IIRL and the expected embedding results. It is assumed that links among node 1, 2, 3, and 4 are structure-close links while links among node 4, 5 and 6 are content-close links. Moreover, the link between node 1 and 3 is also a content-close link. The middle box illustrates the application of IIRL on edge (5,6) and node 5 with its content. Circles in blue denote identity representations while circles in green denote interest representations. a_5 is node attribute vector of node 5, and $||\cdot||_2^2$ is ℓ-2 norm.

is the collection of π_{ij_u}, π_{ij_v}, π_{hk_u} and π_{hk_v}. π_{hk_u} and π_{hk_v} are similar to π_{ij_u}, π_{ij_v} but node h and k are not connected.

By taking negative natural logarithm of Eq. (3), the problem is equivalent to the following minimization problem:

$$\min_{U,V,\pi} -\sum_{e_{ij}\in E} w_{ij}\log p(e_{ij}) - \sum_{e_{hk}\notin E}\log(1-p(e_{hk})) \\ s.t.\ \pi_{ij_u}+\pi_{ij_v}=1\ \&\ \pi_{hk_u}+\pi_{hk_v}=1, \tag{4}$$

where $w_{ij}\in\mathbb{R}$ is the weight of edge e_{ij} to reflect to link strength, and does not violate the original objective as indicated in Eq. (3). The number of e_{hk} can be largely reduced by negative sampling to reduce model complexity as LINE [25] does, and the model performs well with negative sampling as shown in the experiments.

We also illustrate the proposed network structure embedding by a four-layer feedforward neural network model as presented in the middle dashed box of Fig. 3. The input are node representations. By manipulating the connections between the input layer and the first hidden layer, the first hidden layer realizes dimension-wise multiplication of identity representations of two nodes and multiplication of interest representations. The second hidden layer produces identity closeness and interest closeness by employing sigmoid function as the activation function. The output layer produces the overall closeness by summarizing weighted closenesses shown in Eq. (2).

Since user-generated content explicitly indicates user's interests, the interest representation of each node should comply with its content. To achieve this purpose, we regularize interest representations to node content through the following minimization:

$$\min_{V,P}||VP-A||_F^2, \tag{5}$$

where $P\in\mathbb{R}^{D_v\times L}$ is a projection matrix, and $||\cdot||_F^2$ is Frobenius norm. The intuition behind the equation is that the user interests should be well represented by the interest representation V through the projection matrix P. As the first work to explore semantic meanings of social links, the paper assumes the content is nontrivial. Hence, attribute selection is not performed.

By directly combining Eq. (4) for structure embedding and Eq. (5) for content embedding, the overall loss can be drawn as follows:

$$\mathcal{L}(U,V,P,\pi) = -\sum_{e_{ij}\in E} w_{ij}\log p(e_{ij}) - \sum_{e_{hk}\notin E}\log(1-p(e_{hk})) \\ + ||VP-A||_F^2 + \lambda||U||_F^2 + \lambda||V||_F^2 + \lambda||P||_F^2 \\ s.t.\ \pi_{ij_u}+\pi_{ij_v}=1\ \&\ \pi_{hk_u}+\pi_{hk_v}=1, \tag{6}$$

where $||U||_F^2$, $||V||_F^2$, and $||P||_F^2$ are regularization terms, and $\lambda\in\mathbb{R}$ is a regularization coefficient.

Since larger probabilities of edges indicate larger inner products of node representations suggested by Eq. (1), nodes connected by edges are expected to be close in the embedding space. More specifically, nodes connected by structure-close edges should be close in the identity space such as node 1, 2, 3, and 4 in Fig. 3 while nodes connected by content-close edges should be close in the interest space such as node 4 and 5, node 5 and 6. Moreover, node 1 and 3 are close in both identity space and interest space. In this sense, the network structure is well preserved in node representations.

5 THE OPTIMIZATION ALGORITHM

$\mathcal{L}(U,V,P,\pi)$ is not jointly convex over the four input variables. Hence, we replace it with a sequence of easier optimizations by an alternating optimization algorithm [4]. More specifically, the minimization problem can be alternatingly solved with respect to one of the four variables at a time with other variables fixed. We then solve each variable according to the corresponding problem.

Problem (1): The optimization problem for U or V can be solved by gradient-based algorithms, e.g., gradient descent and L-BFGS. The derivative w.r.t U can be obtained as follows:

$$\frac{\partial\mathcal{L}(U,V,P,\pi)}{\partial u_i} = -\sum_{e_{ij}\in E}\frac{w_{ij}\pi_{ij_u}\exp\{-u_i^T u_j\}u_j}{p(e_{ij})(1+\exp\{-u_i^T u_j\})^2} + 2\lambda u_i \\ -\sum_{e_{ik}\notin E}\frac{\pi_{ik_u}\exp\{-u_i^T u_k\}u_k}{(p(e_{ik})-1)(1+\exp\{-u_i^T u_k\})^2}, \tag{7}$$

where e_{ik} is the same concept as e_{hk}.

Problem (2): Similarly, the derivative w.r.t $\boldsymbol{V}$ is as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{P},\boldsymbol{\pi})}{\partial \boldsymbol{v}_i} = &- \sum_{e_{ij}\in E} \frac{w_{ij}\pi_{ij_v}\exp\{-\boldsymbol{v}_i^T\boldsymbol{v}_j\}\boldsymbol{v}_j}{p(e_{ij})(1+\exp\{-\boldsymbol{v}_i^T\boldsymbol{v}_j\})^2} \\ &- \sum_{e_{ik}\notin E} \frac{\pi_{ik_v}\exp\{-\boldsymbol{v}_i^T\boldsymbol{v}_k\}\boldsymbol{v}_k}{(p(e_{ik})-1)(1+\exp\{-\boldsymbol{v}_i^T\boldsymbol{v}_k\})^2} \\ &+ 2(\boldsymbol{v}_i^T\boldsymbol{P} - \boldsymbol{a}_i^T)\boldsymbol{P}^T + 2\lambda\boldsymbol{v}_i\end{aligned} \quad (8)$$

Problem (3): With respect to $\boldsymbol{P}$, the optimization objective turns into solving the following problem:

$$\min_{\boldsymbol{P}} ||\boldsymbol{V}\boldsymbol{P}-\boldsymbol{A}||_F^2 + \lambda||\boldsymbol{P}||_F^2, \quad (9)$$

Hence, the optimal $\boldsymbol{P}$ can be easily obtained by setting the derivative of Eq. (9) with respect to P to zero, where the derivative is quantified as the following formulation:

$$2\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{P} - 2\boldsymbol{V}^T\boldsymbol{A} + 2\lambda\boldsymbol{P}. \quad (10)$$

By setting Eq. (10) to zero, the optimal P is quantified by the following equation:

$$\boldsymbol{P} = (\boldsymbol{V}^T\boldsymbol{V} + \lambda\boldsymbol{I})^{-1}\boldsymbol{V}^T\boldsymbol{A}, \quad (11)$$

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

Problem (4): W.r.t. $\boldsymbol{\pi}$, rather than solving it directly, it is easier to optimize unconstrained "softmax weights" defined as follows:

$$\pi_{ij_u} = \frac{\exp\{\xi_{ij_u}\}}{\exp\{\xi_{ij_u}\} + \exp\{\xi_{ij_v}\}}, \quad (12)$$

where $\xi_{ij_u} \in \mathbb{R}$ and $\xi_{ij_v} \in \mathbb{R}$. As a result, the problem turns into solving ξ_{ij_u} and ξ_{ij_v}, which can be solved by gradient descent.

Hence, the derivative w.r.t ξ_{ij_u} is expressed as follows:

$$\frac{\partial \mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{P},\boldsymbol{\pi})}{\partial \xi_{ij_u}} = \frac{w_{ij}\left[p(\boldsymbol{v}_i,\boldsymbol{v}_j) - p(\boldsymbol{u}_i,\boldsymbol{u}_j)\right]}{p(e_{ij})\left[\exp\{\xi_{ij_u}-\xi_{ij_v}\} + 2 + \exp\{\xi_{ij_v}-\xi_{ij_u}\}\right]}, \quad (13)$$

and $\frac{\partial \mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{P},\boldsymbol{\pi})}{\partial \xi_{ij_v}}$ is similar to $\frac{\partial \mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{P},\boldsymbol{\pi})}{\partial \xi_{ij_u}}$ except that the nominator is $w_{ij}\left[p(\boldsymbol{u}_i,\boldsymbol{u}_j) - p(\boldsymbol{v}_i,\boldsymbol{v}_j)\right]$.

Similarly, the derivatives w.r.t ξ_{hk_u} and w.r.t ξ_{hk_v} are expressed as follows: $\frac{\partial \mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{P},\boldsymbol{\pi})}{\partial \xi_{hk_u}} =$

$$\frac{p(\boldsymbol{v}_h,\boldsymbol{v}_k) - p(\boldsymbol{u}_h,\boldsymbol{u}_k)}{(p(e_{hk})-1)\left[\exp\{\xi_{hk_u}-\xi_{hk_v}\} + 2 + \exp\{\xi_{hk_v}-\xi_{hk_u}\}\right]}, \quad (14)$$

and $\frac{\partial \mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{P},\boldsymbol{\pi})}{\partial \xi_{hk_v}}$ is similar to $\frac{\partial \mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{P},\boldsymbol{\pi})}{\partial \xi_{hk_u}}$ except that the nominator is $p(\boldsymbol{u}_h,\boldsymbol{u}_k) - p(\boldsymbol{v}_h,\boldsymbol{v}_k)$.

The workflow of jointly solving these four variables is presented in Algorithm 1. Algorithm 1 starts with pre-training $\boldsymbol{U}$ and $\boldsymbol{V}$, which is performed to obtain good initialization values for $\boldsymbol{U}$ and $\boldsymbol{V}$ as introduced below, and the parameter k is used in the pre-training. The parameter negative ratio is the ratio of the number of e_{hk} to that of e_{ij} as used in negative sampling by LINE [25]. It is observed that Algorithm 1 works well with this parameter in the experiments. For all gradient descents, the learning rate in each iteration is obtained by backtracking line search [2].

Algorithm 1: The optimization algorithm

```
Input  : $G(N,E,A)$, D, $\xi_{ij_u}$ and $\xi_{ij_v}$, negative ratio, $\lambda$, and $k$
Output : $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{\pi}$
Pre-training $\boldsymbol{U}$ and $\boldsymbol{V}$;
while (not converge) do
    Fixing $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{\pi}$, find the optimal $\boldsymbol{P}$ with the Eq. (11);
    Fixing $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{P}$, find the optimal $\boldsymbol{\pi}$ with gradient descent;
    Fixing $\boldsymbol{V}$, $\boldsymbol{P}$, $\boldsymbol{\pi}$, find the optimal $\boldsymbol{U}$ with gradient descent;
    Fixing $\boldsymbol{U}$, $\boldsymbol{P}$, $\boldsymbol{\pi}$, find the optimal $\boldsymbol{V}$ with gradient descent;
return $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{\pi}$
```

Pre-training is important as it can initialize a model to a point in parameter space that renders the learning process more effective [3]. In our case, to make the learning process more effective, $\boldsymbol{U}$ should be pre-trained to take values reflecting identity-related information as expected, and $\boldsymbol{V}$ should take values reflecting interest-related information. As identity-related information may only be recovered from the network structure given $G(N,E,A)$, we thus pre-train $\boldsymbol{U}$ by embedding the network structure without considering the link type. Based on the proposed network structure embedding, the loss function for pre-training $\boldsymbol{U}$ can be quantified as follows: $L(\boldsymbol{U}) =$

$$-\sum_{e_{ij}\in E} w_{ij}\log(p(\boldsymbol{u}_i,\boldsymbol{u}_j)) - \sum_{e_{hk}\notin E}\log(1-p(\boldsymbol{u}_h,\boldsymbol{u}_k)) + \lambda||\boldsymbol{U}||_F^2, \quad (15)$$

where $p(\boldsymbol{u}_i,\boldsymbol{u}_j)$ and $p(\boldsymbol{u}_h,\boldsymbol{u}_k)$ are quantified in the same way as Eq. (1). We can solve $L(\boldsymbol{U})$ by gradient descent, and the derivative is obtained as follows: $\frac{\partial L(\boldsymbol{U})}{\partial \boldsymbol{u}_i} =$

$$-\sum_{e_{ij}\in E} \frac{w_{ij}\exp\{-\boldsymbol{u}_i^\top\boldsymbol{u}_j\}}{1+\exp\{-\boldsymbol{u}_i^\top\boldsymbol{u}_j\}} \times \boldsymbol{u}_j + \sum_{e_{hk}\notin E} \frac{\boldsymbol{u}_k}{1+\exp\{-\boldsymbol{u}_i^\top\boldsymbol{u}_k\}} + 2\lambda(\boldsymbol{u}_i), \quad (16)$$

As for $\boldsymbol{V}$, because we may not know which links are content-close, we thus pre-train $\boldsymbol{V}$ by embedding node content. To make this problem simple, we construct a k-nearest neighbor graph of the nodes so that the method for pre-training $\boldsymbol{U}$ can be directly applied to pre-train $\boldsymbol{V}$. The links in the kNN graph are established from each node to its first-k neighbors with a weight of 1. The similarity of two nodes is quantified by cosine similarity of their content.

Complexity: Referring to the derivatives, the complexity of Algorithm 1 is dominated by learning node representations from the network structure and the term frequency matrix constructed from node content. Taking the identity representations as an example, the complexity of learning from the network structure is $O(D_u|E|)$ due to negative sampling and the complexity of learning from the term frequency matrix is $O(D_u|N|L)$ where L is the number of terms. Since social networks are usually sparse, the scalability to large-scale networks can be guaranteed.

Convergence: Algorithm 1 is essentially a block-wise coordinate descent algorithm [27] with $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{\pi}$ and $\boldsymbol{P}$ as block variables. So convergence can be guaranteed based on the general proof of convergence for block-wise coordinate descent. Moreover, in the experiments, we observe Algorithm 1 converges fast in terms of the outer iterations.

Network	DBLP	Twitter	BlogCatalog	Flickr
#Nodes	6482	9244	7857	6318
#Edges	19265	323922	137649	404085
#Attributes	8298	10360	5351	8523
#Groups	4	N/A	15	5
Type	weighted undirected	unweighted directed	unweighted undirected	unweighted undirected

Table 1: Network statistics.

6 EMPIRICAL EVALUATION

6.1 Datasets

Four social networks studied in the experiments are as follows:

- DBLP [26]: We sample a co-authorship network of authors who published as least 2 papers during the period from 2000 to 2009 from selected conferences, which are KDD, ICDM, SDM, and PAKDD in DM field, AAAI, ICML, NIPS, IJCAI, CVPR, and ECML in ML field, SIGMOD, VLDB, ICDE, PODS, and EDBT in DB field, WWW, SIGIR, CIKM, WSDM, and ECIR in IR field. Words of paper abstract are assigned as the attributes of each author. Stop words and words with frequency less than 6 are omitted.
- Twitter [17]: The dataset has directed links among users, and words from tweets demonstrating user's interests. We sample users who have attributes no less than 20 and have followed no less than 6 users, and sample the words with a minimal frequency of 8.
- BlogCatalog [32]: The links among users are undirected friendships. We aggregate tags on blogs as users' attributes. We sample users who published as least one blog belonging to one of 15 popular categories, which are Art, Computers, Music, Photography, Travel, Sports, Technology, Internet, Humor, Business, Writing, Culture, Health, Finance, and Film. The threshold of attributes for each user is set as 6, and the threshold of the number of friends is 3. Attributes with frequency less than 3 are omitted.
- Flickr[31]: The links between users are friendships, and the attributes for each user are aggregated tags on their photos. We sample users from five popular groups which are 7, 36, 77, 119, and 148. We sample users with minimal 30 attributes and with minimal 10 friends, and set the threshold of frequency for each feature as 25.

Network statistics are summarized in Table 1.

6.2 Experiment Settings

Five recent network embedding models are employed as baselines, which are DeepWalk [21], LINE [25], and node2vec [11] for embedding only the network structure, TADW [40] and EOE [37] for embedding both the network structure and node content.

In the experiments, the dimension of both identity representation and interest representation is set as 128 as used in all the baselines. Because there are two representations for each node in both IIRL and TADW, the two representations are concatenated as one representation in the evaluation except for the representations of IIRL used in link prediction. Though the dimension of node representation of both IIRL and TADW is two times of that of other baselines, a larger dimension actually does not bring advantages as suggested in the parameter sensitivity of this paper, of LINE, of TADW, and of EOE. Other settings are that ξ_{ij_u} and ξ_{ij_v} are initialized as 0.5, k for pre-training is set as top 1% of all the nodes, negative ratio is set as 5 as used in LINE, the regularization coefficient is set as 1, commonly used settings are used in backtracking line search, and the relative loss that determines the convergence of Algorithm 1 is set as 0.001.

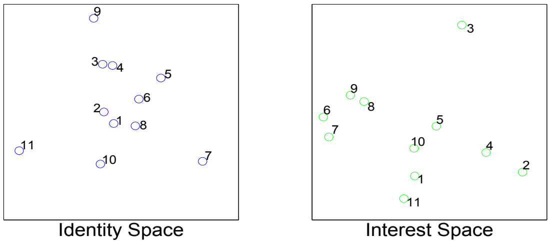

Figure 4: Locations of Twitter nodes in different spaces.

6.3 Representation Visualization

This section presents a visual evaluation of the effectiveness of the identity representations and interest embeddings by visualizing them in a two-dimension space. Firstly, we present locations of 11 nodes from the Twitter network in identity space and interest space in Fig. 4, which is obtained by t-SNE [16]. The true user IDs in Twitter for these 11 nodes are listed in the following Table 4. The relationships between them are that 1&2, 3&4, and 5&6 have many common friends, but have shown little or no similar interests. This may suggest that links between them may be purely structure-close. 6&7, 8&9, and 10&11 show similar interests but no or one common neighbor, which may suggest the interest proportion of these links is larger than identity proportion. Moreover, node 5, 6 and 7 is a real example of the motivation scenario in Fig. 1 in the introduction. As suggested by the model illustration in Fig. 3, 1&2, 3&4 and 5&6 should be close in the identity space and far away in the interest space while 6&7, 8&9, and 10&11 should have the exactly opposite pattern. Moreover, node 7 should be far away from node 5 in both spaces even though they share node 6 as a common neighbor. By examining Fig. 4, we find the representations work as expected.

Secondly, we visualize representations in network scale. The DBLP co-authorship network is studied as it is a well-structured professional network. The results are presented in Fig.5.

Co-authorships tend to take place within each research field, but the selected four fields, i.e., DB, DM, ML, and IR, are closely related. In other words, many co-authorships may be cross-filed, which implies that the co-authorships alone may not be sufficient to distinguish researchers from different fields. Hence, representations learned from the network structure may not be able to visually distinguish authors from one field to another. And this is indeed demonstrated by all the models that can only embed the network structure. But TADW, EOE, and the proposed IIRL work much better by combining information of the network structure and of the node content as illustrated in Fig. 5(5), Fig. 5(6), and Fig. 5(7), respectively. This is because node content demonstrates research interests, which are different among these four fields as illustrated in Fig. 5(8) and Fig. 5(9).

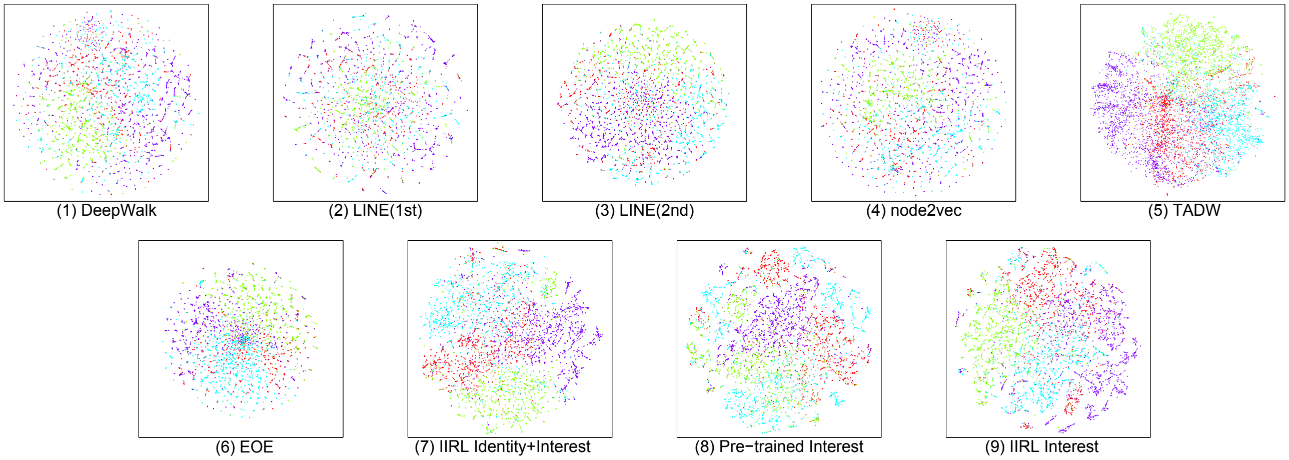
(1) DeepWalk (2) LINE(1st) (3) LINE(2nd) (4) node2vec (5) TADW
(6) EOE (7) IIRL Identity+Interest (8) Pre−trained Interest (9) IIRL Interest

Figure 5: Visualization of representations learned from the DBLP co-authorship network. Points of different colors denote authors from different research fields, where red color represents Data Mining, green color represents Database, light blue represents Information Retrieval, and dark blue represents Machine Learning. The filed of each author is determined as the one where he/she published the most papers.

The proposed IIRL and TADW perform a slightly better than EOE, which may because research interests are directly combined with representations learned from the network structure in IIRL and TADW while EOE can only incorporate interests in an implicit way. Though it is not easy to visually tell which one of IIRL and TADW performs better, the way of IIRL to combine the network structure and node content is more reasonable. TADW performs the combination by approximating the entire network structure through node content. Analogously, IIRL performs the approximation selectively, which is motivated by the fact that different edges have different proportions of interest characteristics and identity characteristics. IIRL performs the selection through inferring responsibility weights so that node characteristics can be embedded in appropriate representations other than are blindly embedded in identity representations or interest ones. The effectiveness of this selective embedding is demonstrated by the observation that final interest representations in Fig. 5(9) perform better than pre-trained ones in Fig. 5(8). The performance of how correctly IIRL infers responsibility weights is evaluated via link type inference below.

6.4 Link Type Inference

As there is no ground truth about the identity proportion and interest proportion of each edge given $G(N, E, A)$, we define four types of links in terms of common neighbors and the relationship in interests in Table 2. The neighbor relationship is defined in the pre-training section. Type *I* corresponds to purely structure-close social links, Type *II* corresponds to purely content-close social links, and Type *III* corresponds to both structure-close and content-close links while Type *IV* remains uncertain as there is no clear clue about the type of node characteristics. Bottom of Table 1 lists the statistics of four types of links of the Twitter network.

It is worthy of noting that the distribution of Twitter links over the four types depends on the choice of the number k in the k-nearest neighbor network. The number k determines the number of links composed of nodes with neighbor relationship in interests. Here, the k is set as 1% of nodes, a commonly used number for constructing kNN networks in network embedding [22]. As a result, the number of neighbor links is 109320, taking up 33.7% of all the links. The number is reasonable as there more than 60% of links whose endpoint vertices share few interests as indicated by small similarities (e.g., ≤ 0.2) in the cumulative density plot of Fig. 2.

	Type I	Type II	Type III	Type IV
Sharing neighbors	✓	✗	✓	✗
Not sharing neighbors	✗	✓	✗	✓
Neighbor in interests	✗	✓	✓	✗
Non-neighbor in interests	✓	✗	✗	✓
Twitter	208598	845	108475	6014

Table 2: Four types of social links in terms of the common neighbors and the relationship in interests.

	Predicted Type *I*	Predicted Type *II*	Recall Rate
True Type *I*	181751	26847	87.13%
True Type *II*	640	205	75.73%
Precision	99.65%	99.24%	

Table 3: Confusion matrix for link type inference.

We can quantitatively evaluate how IIRL correctly infers Type *I* and Type *II* links in a way that Type *I* links should have larger values of π_{ij_u} while Type *II* links should have larger values of π_{ij_v}. For Type *III* links, we may not be able to do a quantitative evaluation but we can do a qualitative evaluation based on the

Types	User ID (R. N.)	#C. F.	N. I.	Common Attributes	π_{ij_u}	π_{ij_v}
Type *I*	4119741 (1), 32423136 (2)	22	✗	@TheAtlantic	0.82	0.18
	155969606 (3), 65913144 (4)	23	✗	N/A	0.89	0.11
	618593 (5), 157218534 (6)	9	✗	N/A	0.87	0.13
Type *II*	157218534 (6), 387210978 (7)	0	✓	#StarWars, @darthvader, @DepressedDarth	0.15	0.85
	85815410 (8), 23508439 (9)	1	✓	#Cowboys, #Patriots, #NFLDraft	0.12	0.88
	11348282 (10), 77351627 (11)	1	✓	#gamedev, @jonjones, @polycount, @GearboxSoftware	0.28	0.72

Table 4: Representatives of Type *I* and Type *II* links, where R. N. is acronym of Reference Number used in Section 6.3, #C. F. is acronym of #Common Friends, and N. I. is acronym of Neighbor in Interests.

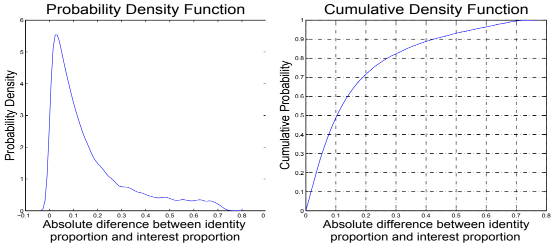

Figure 6: PDF plot and CDF plot

difference between π_{ij_u} and π_{ij_v}, which should not be too large because there may not be a dominant factor for Type *III* links.

The confusion matrix on inferring Type *I* and Type *II* social links is presented in Table 3. It shows both Type *I* and Type *II* links can be accurately inferred, and the performance on inferring Type *I* links is better. To verify this performance, three representatives of accurately predicted Type *I* links and three of Type *II* links are presented in Table 4. We can see from Table 4 that all estimated π_{ij_u} and π_{ij_v} are consistent with link types. Also, for Type *I* links, besides there are many common friends shared by the two users, the friendship is bi-directional, which largely supports the claim that two users connected by structure-close links may have an idea of whom each other is in the real world.

To evaluate the performance on inferring Type *III* links, we present the distribution of absolute differences between π_{ij_u} and π_{ij_v} and the cumulative density plot in Fig. 6. The PDF plot shows the majority of differences are small, and the CDF plot shows around 70% of differences are within 0.2. Because it is expected that Type *III* links may not have a dominant type of characteristics, the identity proportions should take values similar to that of the interest proportions. To this point, it is concluded that the proposed IIRL model can accurately infer link types, and responsibility weights.

6.5 Social Profiling

After inferring link types, we can build *social profiles* so as to understand users' social behaviors and the social network. The social profile of a user refers to information about his/her friendships with other users. Social profiling is a by-product of learning representations because IIRL is able to infer the type of social links of which all the baselines, in contrast, are not capable.

According to the link types, we categorize users into five groups as illustrated in Table 5, where *only identity-exposing* users are those who only have structure-close links, and other terms are defined similarly. It is worthy of noting that structure-close links here are inferred by IIRL during the training process, and they are associated with relatively larger identity proportion π_{ij_u} compared with interest proportion π_{ij_v}. Social profiles in network scale by summarizing over users are also presented in Table 5. Two interesting observations are drawn as follows:

		Twitter	Flickr	Blog	DBLP
#Users	Only identity-exposing	240	1359	403	1378
	Only interest-exposing	1837	1170	79	1937
	More Identity-exposing	3389	3876	5120	1486
	More Interest-exposing	3595	1227	590	1642
	Equally exposing	116	259	126	39
#Links	structure-close	218158	125220	352531	21568
	content-close	105764	12429	53157	16962

Table 5: Social profiling for networks.

- From the perspective of links, all networks show that more social links are established between users who may know each other in the real world because structure-close links out-number content-close links. This suggests that the major reason for people to involve in social networks may be to maintain relationships with real-world friends. Also, all the networks except the Twitter network agree with the conclusion from the perspective of users. The reason behind the Twitter network is explored below.
- Uni-directional relationships would encourage connections between people who do not know each other. By referring to the Twitter network, there are more interest-exposing users, and the ratio of content-close links to structure-close links is not that small compared with that in Flickr and BlogCatalog. This is excepted because friendships are easier to be acknowledged by one side than by both sides.

6.6 Link Prediction

Link prediction is usually performed by measuring similarities of pairs of nodes [15]. Inner product of two node representations normalized by sigmoid function is employed as the similarity measurement for all the baselines. For IIRL, the similarity is the summary of identity similarity and interest similarity weighted by identity proportion and interest proportion. On Twitter, BlogCatalog, and Flickr networks, we conduct 4 runs of experiments in which different ratios of links are used as training links and the remaining ones as test

Models	BlogCatalog				Flickr				Twitter				DBLP
	20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%	100%
DeepWalk	69.92	76.44	79.32	81.32	71.79	77.04	78.30	78.69	96.25	97.62	97.68	97.80	73.26
TADW	67.12	70.67	73.02	74.97	62.59	65.04	68.10	70.16	94.25	95.24	95.82	96.90	66.26
LINE(1st)	53.61	73.70	65.36	79.60	61.31	66.31	69.10	70.83	92.56	95.90	97.02	97.70	62.45
LINE(2nd)	65.77	68.25	70.47	72.05	66.25	68.02	68.66	68.79	91.04	94.84	96.12	96.63	76.04
node2vec	73.69	76.99	77.75	78.54	71.20	70.80	70.71	71.75	96.12	97.64	97.68	97.68	71.59
EOE	78.28	83.36	86.81	88.94	74.10	77.36	78.69	79.39	96.93	97.79	98.01	98.09	76.19
IIRL	**82.30**	**86.54**	**90.26**	**91.92**	**78.62**	**79.41**	**82.03**	**81.29**	**97.60**	**98.69**	**98.37**	**98.45**	**77.21**

Table 6: AUC scores of link prediction tasks, where percentage numbers are percentages of links used as training links.

Models	Micro-F1			Macro-F1		
	Blog	Flickr	DBLP	Blog	Flickr	DBLP
DeepWalk	58.42	66.52	78.48	57.86	55.71	77.46
TADW	63.62	67.96	84.68	61.55	58.58	83.96
LINE(1st)	59.55	63.95	77.62	58.72	56.92	76.51
LINE(2nd)	57.60	66.07	75.93	56.95	56.55	75.45
node2vec	55.85	57.76	76.96	55.73	48.72	75.66
EOE	62.95	67.86	84.56	62.03	58.62	83.32
IIRL	**66.08**	**68.79**	**85.30**	**63.53**	**59.36**	**84.77**

Table 7: Micro-F1 and Macro-F1 of multi-label classification

links. On the DBLP co-authorship network, new co-authorships occur from 2010 to 2012 are used as test links. For all networks, the same number of negative links are randomly sampled for the evaluation purpose. For IIRL, if a particular pair of nodes does not appear during the training process, the identity proportion and interest proportion are both set as initialized 0.5. The performance measured on AUC score is presented in Table 6.

Table 6 shows the proposed IIRL model consistently outperforms all baselines on all datasets. The reason behind the superior performance of IIRL over DeepWalk, LINE and node2vec is that a considerable number of links should be content-close as demonstrated previously and IIRL can capture users' interests exhibited in user-generated content. The superior performance over TADW is because not all links are content-close. Moreover, TADW even under-performs DeepWalk except on the Twitter network which has more content-close links than the other three networks. Hence, it not appropriate to approximate the entire network structure by node content like TADW but to approximate the content-close links by node content like the proposed IIRL. The superior performance over EOE is because interests are more explicitly employed in inferring new interactions. Moreover, the unexpected consequence of learning a single representation for two types of node characteristics mentioned in the introduction may also explain the inferior performance of all the baselines.

Interpretability on link prediction: Besides being more accurate, IIRL can provide explanations for each piece of link prediction, e.g., it is established because two users know each other in the real world or because two users share similar interests. This can be easily obtained by referring to identity similarity and interest similarity because the larger similarity is expected to be associated with the larger proportion as discussed in Definition 4.

6.7 Multi-label Classification

The groups are used as labels for the evaluation. We employ the binary-relevance SVM with polynomial kernel as the classifier, and the performance in terms of Micro-F1 and Macro-F1 obtained by 5-fold cross validation is presented in Table 7. Similarly, IIRL outperforms all the baselines. An intuitive explanation for the superior performance can be obtained from the visualization of the DBLP network in Fig. 5. Specifically, as nodes of the same group are distributed closer to each other in IIRL than in baselines, the decision boundary estimated in SVM can be more effective.

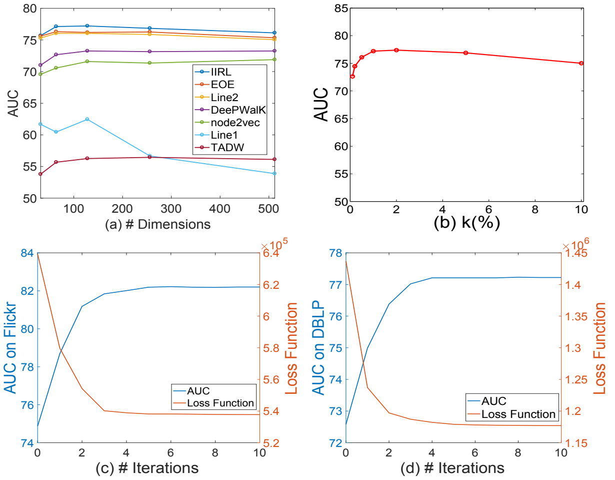

Figure 7: Sensitivity analysis and convergence analysis

6.8 Parameter Sensitivity and Convergence

In this section, we study the performance w.r.t to the dimension of representations and the number k used in kNN networks. Fig. 7(a) presents the performance on the link prediction task for the DBLP dataset w.r.t different dimensions of representations. It shows that the proposed IIRL is not much sensitive to the dimension. Fig. 7(b) presents the performance on the link prediction task for the DBLP dataset w.r.t k. It shows that the number k should not be too small (e.g., 0.1%) as well as too large (e.g., 10%).

We also study the convergence of Algorithm 1. Specifically, we study the performance of the algorithm on link prediction for Flickr when 80% of links used as training data and for DBLP w.r.t. the number of outer iterations. The results are presented in Fig. 7 (c) and Fig. 7 (d). It shows that the algorithm converges very fast and may converge to stable performance after about 5 outer iterations.

7 CONCLUSION AND FUTURE WORK

In this paper, we explore semantic meanings of social links in network embedding for the first time. We categorize social links according to whether they are structure-close or content-close, and propose IIRL to learn identity representations and interest representations. In the future, we plan to explore more fine-grained semantic meanings of social links instead of two coarse-grained meanings.

ACKNOWLEDGEMENT

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 University's Support for Application of Major Research Funding provided by The Hong Kong Polytechnic University (Project Code: 1-BBA1), RGC General Research Fund under Grant PolyU 152199/17E, NSFC Key Grant with Project No. 61332004, NSF through grants IIS-1526499, and CNS-1626432, and NSFC 61672313.

REFERENCES

[1] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. 2006. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*.

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

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

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

[5] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. 2011. Node classification in social networks. In *Social network data analytics*. Springer, 115–148.

[6] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 891–900.

[7] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. 2015. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 119–128.

[8] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2172–2180.

[9] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.

[10] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3 (2010), 75–174.

[11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[12] William H Hsu, Andrew L King, Martin SR Paradesi, Tejaswi Pydimarri, and Tim Weninger. 2006. Collaborative and Structural Recommendation of Friends using Weblog-based Social Network Analysis.. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, Vol. 6. 55–60.

[13] Juzheng Li, Jun Zhu, and Bo Zhang. 2016. Discriminative Deep Random Walk for Network Classification.. In *ACL (1)*.

[14] Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, and Kevin Chen-Chuan Chang. 2012. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1023–1031.

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

[16] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[17] Julian J McAuley and Jure Leskovec. 2012. Learning to Discover Social Circles in Ego Networks.. In *NIPS*, Vol. 2012. 548–56.

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

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[20] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric Transitivity Preserving Graph Embedding.. In *KDD*. 1105–1114.

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

[22] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 5500 (2000), 2323–2326.

[23] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S Yu. 2017. Heterogeneous Information Network Embedding for Recommendation. *arXiv preprint arXiv:1711.10730* (2017).

[24] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1165–1174.

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

[26] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 990–998.

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

[28] Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, and Maosong Sun. 2016. Max-Margin DeepWalk: Discriminative Learning of Network Representation.. In *IJCAI*. 3889–3895.

[29] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1225–1234.

[30] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community Preserving Network Embedding.. In *AAAI*. 203–209.

[31] Xufei Wang, Lei Tang, Huiji Gao, and Huan Liu. 2010. Discovering Overlapping Groups in Social Media. In *the 10th IEEE International Conference on Data Mining series (ICDM2010)*. Sydney, Australia.

[32] Xufei Wang, Lei Tang, Huan Liu, and Lei Wang. 2012. Learning with Multi-Resolution Overlapping Communities. *Knowledge and Information Systems (KAIS)* (2012). https://doi.org/10.1007/s10115-012-0555-0

[33] Xiaokai Wei, Bokai Cao, and S Yu Philip. 2016. Unsupervised Feature Selection on Networks: A Generative View.. In *AAAI*. 2215–2221.

[34] Xiaokai Wei, Sihong Xie, and Philip S Yu. 2015. Efficient partial order preserving unsupervised feature selection on networks. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 82–90.

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

[36] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S Yu. 2017. Disentangled Link Prediction for Signed Social Networks via Disentangled Representation Learning. In *Proceedings of the Fourth IEEE International Conference on Data Science and Advanced Analytics*. IEEE.

[37] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S Yu. 2017. Embedding of embedding (eoe): Joint embedding for coupled heterogeneous networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 741–749.

[38] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S Yu. 2017. Multi-task Network Embedding. In *Proceedings of the Fourth IEEE International Conference on Data Science and Advanced Analytics*. IEEE.

[39] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S Yu. 2017. Multiple Social Role Embedding. In *Proceedings of the Fourth IEEE International Conference on Data Science and Advanced Analytics*. IEEE.

[40] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network representation learning with rich text information. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina*. 2111–2117.

[41] Cheng Yang, Maosong Sun, Zhiyuan Liu, and Cunchao Tu. 2017. Fast network embedding enhancement via high order proximity approximation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*. 19–25.

[42] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861* (2016).

[43] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 283–292.