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

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

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

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

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 sparse. 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

KEYWORDS
Network embedding; social networks; data mining
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).

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 transivities 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-related characteristics, 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:

$$p(e_{iju}) = \frac{1}{1 + \exp(-u_i^T u_j)}$$

where $e_{iju}$ is a structure-close edge, $u_i \in \mathbb{R}^{D_u}$ and $u_j \in \mathbb{R}^{D_u}$ 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:

$$p(e_{ij}) = \pi_{iju} p(e_{iju}) + \pi_{ijv} p(e_{ijv})$$

where $p(e_{iju})$ quantifies interest closeness, $e_{iju}$ is a content-close edge, and $\pi_{iju}$ and $\pi_{ijv}$ are defined as follows:

**DEFINITION 3.** Identity proportion of a linkage relationship denoted as $\pi_{iju}$ 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 $\pi_{ijv}$ quantifies the responsibility weight of interest-related characteristics. As there are only two types of node characteristics as indicated in the introduction, $\pi_{iju} + \pi_{ijv} = 1$.

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:

$$\max_{U, V, \pi} \prod_{e_{ij} \in E} p(e_{ij}) \prod_{e_{hk} \in E} (1 - p(e_{hk}))$$

s.t. $\pi_{iju} + \pi_{ijv} = 1$ and $\pi_{hku} + \pi_{hkv} = 1$,

where $U \in \mathbb{R}^{M \times D_u}$ and $V \in \mathbb{R}^{M \times D_v}$ is the matrix of identity representations and interest representations, respectively, and $\pi$
is the collection of \( \pi_{ij_a}, \pi_{ij_v}, \pi_{hk_u} \) and \( \pi_{hk_v} \), and \( \pi_{hk_v} \) are similar to \( \pi_{ij_a}, \pi_{ij_v} \) 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:

\[
\min_{U, V, \pi} \sum_{\epsilon_{ij} \in E} w_{ij} \log p(\epsilon_{ij}) - \sum_{\epsilon_{hk} \not\in E} \log(1 - p(\epsilon_{hk})) \\
\text{s.t. } \pi_{ij_a} + \pi_{ij_v} = 1 \text{ and } \pi_{hk_u} + \pi_{hk_v} = 1,
\]

where \( w_{ij} \in \mathbb{R} \) is the weight of edge \( \epsilon_{ij} \) to reflect to link strength, and does not violate the original objective as indicated in Eq. (3).

The number of \( \epsilon_{hk} \) 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:

\[
\min_{V, P} ||VP - A||^2_F,
\]

where \( P \in \mathbb{R}^{D_u \times L} \) is a projection matrix, and \( || \cdot ||^2_F \) 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 non-trivial. 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:

\[
\mathcal{L}(U, V, \pi) = -\sum_{\epsilon_{ij} \in E} w_{ij} \log p(\epsilon_{ij}) - \sum_{\epsilon_{hk} \not\in E} \log(1 - p(\epsilon_{hk})) \\
+ ||VP - A||^2_F + \lambda(||U||^2_F + ||V||^2_F + ||\pi||^2_F) \\
\text{s.t. } \pi_{ij_a} + \pi_{ij_v} = 1 \text{ and } \pi_{hk_u} + \pi_{hk_v} = 1,
\]

where \( ||U||^2_F, ||V||^2_F, \) and \( ||\pi||^2_F \) 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

\( \mathcal{L}(U, V, \pi) \) 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:

\[
\frac{\partial \mathcal{L}(U, V, \pi)}{\partial u_i} = -\sum_{\epsilon_{ij} \in E} w_{ij} \pi_{ij_a} \exp[-u_j^T u_i] - \lambda u_i \\
- \sum_{\epsilon_{hk} \not\in E} (p(\epsilon_{hk}) - 1)(1 + \exp[-u_h^T u_k]) u_k,
\]

where \( \mathcal{L}(U, V, \pi) \) is the collection of identity representations while circles in green denote interest representations. a\( _5 \) is node attribute vector of node 5, and \( || \cdot ||^2_F \) is \( \ell^2-2 \) norm.
where $e_{ik}$ is the same concept as $e_{hk}$.

**Problem (2):** Similarly, the derivative w.r.t $V$ is as follows:

$$
\frac{\partial \mathcal{L}(U, V, P, \pi)}{\partial u_i} = -\sum_{e_{ij} \in E} \frac{w_{ij} \pi_{ij}}{p(e_{ij})} \exp(-w^T u_j) v_j
- \sum_{e_{ik} \in E} \frac{\pi_{ik}}{p(e_{ik})} \exp(-w^T u_k) v_k
+ (w^T u + d_i)^T P^T + 2\alpha v_i
$$

**Problem (3):** With respect to $P$, the optimization objective turns into solving the following problem:

$$
\min_P \|VP - A\|^2_F + \lambda \|P\|^2_F,
$$

Hence, the optimal $P$ 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:

$$
2VTVP - 2VT A + 2\lambda P = 0.
$$

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

$$
P = (VT^2 + \lambda I)^{-1} VT A,
$$

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

**Problem (4):** W.r.t. $\pi$, rather than solving it directly, it is easier to optimize unconstrained 'softmax weights' defined as follows:

$$
\pi_{ij} = \frac{\exp(\xi_{ij})}{\exp(\xi_{ij}) + \exp(\xi_{ji})},
$$

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

Hence, the derivative w.r.t $\xi_{ij}$ is expressed as follows:

$$
\frac{\partial \mathcal{L}(U, V, P, \pi)}{\partial \xi_{ij}} = \frac{w_{ij} \left[p(u_i, u_j) - p(u_i, u_j)\right]}{p(e_{ij}) \left[\exp(\xi_{ij} - \xi_{ji}) + 2 + \exp(\xi_{ij} - \xi_{ji})\right]},
$$

and $\frac{\partial \mathcal{L}(U, V, P, \pi)}{\partial \xi_{ji}}$ is similar to $\frac{\partial \mathcal{L}(U, V, P, \pi)}{\partial \xi_{ij}}$ except that the numerator is $w_{ij} \left[p(u_i, u_j) - p(u_i, u_j)\right]$.

Similarly, the derivatives w.r.t $\xi_{hk}$ and w.r.t $\xi_{kh}$, are expressed as follows:

$$
\frac{\partial \mathcal{L}(U, V, P, \pi)}{\partial \xi_{hk}} = \frac{p(u_h, u_k) - p(u_h, u_k)}{(p(e_{hk}) - 1) \left[\exp(\xi_{hk} - \xi_{kh}) + 2 + \exp(\xi_{hk} - \xi_{kh})\right]},
$$

and $\frac{\partial \mathcal{L}(U, V, P, \pi)}{\partial \xi_{kh}}$ is similar to $\frac{\partial \mathcal{L}(U, V, P, \pi)}{\partial \xi_{hk}}$ except that the numerator is $p(u_h, u_k) - p(u_h, u_k)$.

The work of jointly solving these four variables is presented in Algorithm 1. Algorithm 1 starts with pre-training $U$ and $V$, which is performed to obtain good initialization values for $U$ and $V$ 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:** $G(N, E, A)$, $\xi_{ij}$ and $\xi_{ij}$, negative ratio, $\lambda$, and $k$

**Output:** $U$, $V$, and $\pi$

Pre-training $U$ and $V$;

while (not converge) do

  Fixing $U$, $V$, $\pi$, find the optimal $P$ with the Eq. (11);
  Fixing $U$, $V$, $P$, find the optimal $\pi$ with gradient descent;
  Fixing $V$, $P$, $\pi$, find the optimal $U$ with gradient descent;
  Fixing $U$, $P$, $\pi$, find the optimal $V$ with gradient descent;

return $U$, $V$, and $\pi$

**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, $U$ should be pre-trained to take values reflecting identity-related information as expected, and $V$ 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 $U$ by embedding the network structure without considering the link type. Based on the proposed network structure embedding, the loss function for pre-training $U$ can be quantified as follows:

$$
L(U) = -\sum_{e_{ij} \in E} w_{ij} \log(p(u_i, u_j)) - \sum_{e_{hk} \notin E} \log(1 - p(u_h, u_k)) + \lambda \|U\|^2_F,
$$

where $p(u_i, u_j)$ and $p(u_h, u_k)$ are quantified in the same way as Eq. (1). We can solve $L(U)$ by gradient descent, and the derivative is obtained as follows:

$$
\frac{\partial L(U)}{\partial u_i} = \\
-\sum_{e_{ij} \in E} \frac{w_{ij}}{1 + \exp[-u^T u_j]} u_j + \sum_{e_{hk} \notin E} \frac{u_h}{1 + \exp[-u^T u_k]} + 2\lambda u_i.
$$

As for $V$, because we may not know which links are content-wise, we thus pre-train $V$ 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 $U$ can be directly applied to pre-train $V$. 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 $O(D^2 a |E|)$ due to negative sampling and the complexity of learning from the term frequency matrix is $O(D a |N| |L|)$ 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 $U$, $V$, $\pi$ and $P$ 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.
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 PARADISE 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.

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

Table 1: Network statistics.

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 $\xi_{ij}$ and $\xi_{ij}$ 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.

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 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(8) and Fig. 5(9).
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 difference 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 social 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., \( \leq 0.2 \)) in the cumulative density plot of Fig. 2.

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 \( \pi_{ij} \) while Type II links should have larger values of \( \pi_{iji} \). For Type III links, we may not be able to do a quantitative evaluation but we can do a qualitative evaluation based on the

![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.](image-url)

<table>
<thead>
<tr>
<th>Sharing neighbors</th>
<th>Type I</th>
<th>Type II</th>
<th>Type III</th>
<th>Type IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not sharing neighbors</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Neighbor in interests</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Non-neighbor in interests</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Predicted Type I</th>
<th>Predicted Type II</th>
<th>Recall Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Type I</td>
<td>181751</td>
<td>26847</td>
<td>87.13%</td>
</tr>
<tr>
<td>True Type II</td>
<td>640</td>
<td>205</td>
<td>75.73%</td>
</tr>
<tr>
<td>Precision</td>
<td>99.65%</td>
<td>99.24%</td>
<td></td>
</tr>
</tbody>
</table>

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

Table 3: Confusion matrix for link type inference.
difference between \( \pi_{ij_u} \) and \( \pi_{ij_v} \), 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 \( \pi_{ij_u} \) and \( \pi_{ij_v} \) 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 \( \pi_{ij_u} \) and \( \pi_{ij_v} \) 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 \( \pi_{ij_u} \) compared with interest proportion \( \pi_{ij_v} \). Social profiles in network scale by summarizing over users are also presented in Table 5. Two interesting observations are drawn as follows:

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

<table>
<thead>
<tr>
<th>Models</th>
<th>Blog</th>
<th>Flickr</th>
<th>DBLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>69.92</td>
<td>76.44</td>
<td>79.32</td>
</tr>
<tr>
<td>TADW</td>
<td>67.12</td>
<td>70.67</td>
<td>73.02</td>
</tr>
<tr>
<td>LINE(1st)</td>
<td>53.61</td>
<td>73.70</td>
<td>65.36</td>
</tr>
<tr>
<td>LINE(2nd)</td>
<td>65.77</td>
<td>68.25</td>
<td>70.47</td>
</tr>
<tr>
<td>node2vec</td>
<td>73.69</td>
<td>76.99</td>
<td>77.75</td>
</tr>
<tr>
<td>EOE</td>
<td>78.28</td>
<td>83.36</td>
<td>86.81</td>
</tr>
<tr>
<td>IIRL</td>
<td>82.30</td>
<td>86.54</td>
<td>90.26</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Models</th>
<th>Blog</th>
<th>Flickr</th>
<th>DBLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>58.42</td>
<td>66.52</td>
<td>78.48</td>
</tr>
<tr>
<td>TADW</td>
<td>63.62</td>
<td>67.96</td>
<td>84.68</td>
</tr>
<tr>
<td>LINE(1st)</td>
<td>59.55</td>
<td>63.95</td>
<td>77.62</td>
</tr>
<tr>
<td>LINE(2nd)</td>
<td>57.60</td>
<td>66.07</td>
<td>75.93</td>
</tr>
<tr>
<td>node2vec</td>
<td>55.85</td>
<td>57.76</td>
<td>76.96</td>
</tr>
<tr>
<td>EOE</td>
<td>62.95</td>
<td>67.86</td>
<td>84.56</td>
</tr>
<tr>
<td>IIRL</td>
<td>66.08</td>
<td>68.79</td>
<td>85.30</td>
</tr>
</tbody>
</table>

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.

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.
propose IRL 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