

# Structural Representation Learning for User Alignment Across Social Networks

Li Liu, Xin Li *Member, IEEE*, William K. Cheung, *Member, IEEE*, and Lejian Liao

**Abstract**—Aligning users across different social networks has become increasingly studied as an important task to social network analysis. In this paper, we propose a novel representation learning method that mainly exploits social structures for the network alignment. In particular, the proposed network embedding framework models the follower-ship and followee-ship of each user explicitly as input and output context vectors, while preserving the proximity of users with “similar” followers and followees in the embedded space. We incorporate both known and predicted user anchors across the networks as constraints to facilitate the transfer of context information to achieve accurate user alignment. Both network embedding and user alignment are inferred under a unified optimization framework with negative sampling adopted to ensure scalability. Also, variants of the proposed framework, including the incorporation of higher-order structural features, are also explored for further boosting the alignment accuracy. Extensive experiments on large-scale social and academia network datasets demonstrate the efficacy of our proposed model compared with state-of-the-art methods.

**Index Terms**—User Alignment, Network Embedding, Representation Learning, Social Networks.

## 1 INTRODUCTION

PARTICIPATING in multiple online social networks to share and communicate has become a new culture in modern life. People sign up different social media for different purposes. Even though many social network services support the use of shared accounts for login, it is still common that users manage multiple social media accounts. More recently, the challenge of aligning users across online social networks has attracted attention in both academia and industry. The established user correspondence can support applications such as preferred link prediction [1], [2], [3] and cross-domain recommendation [4], [5]. In general, aligning users across multiple social networks can in turn help alleviate the sparsity issue of analyzing individual social networks with information fusion.

One intuitive way to establish the user correspondence is to base on user attributes, including username, gender, and other demographic information. While the user attribute based approach has been demonstrated effective under some given experimental settings [6], [7], [8], it fails in a number of scenarios. For instance, there exist quite a lot of users who deliberately use different usernames. Also, demographic information in different networks is highly likely to be unbalanced, and the presence of rich and correct profiles sometimes cannot be always assumed. Other than user attributes, some also proposed to map users using their long-term topical interest, language style of personalized wordings, and emoticon adoption [9], [10].

Alternatively, structural information of the social networks can be used directly for user alignment. In particular, intra-links within a social network and inter-links across dif-

ferent networks (also called anchor users) can be exploited to, say, derive a probabilistic graph classifier [11], [12], or to render a common subspace of multiple networks for relevance computation [12], [13]. While promising results have been reported, there still exist a number of limitations. First, most of the related work considers the links to be undirected. The fact is that follower-followee relations are maintained in some media like Twitter and are better to be represented as directed graphs as what being considered in this paper. In general, the conformation of the follower-ship of a user somehow reflects the objective recognition from the community, whereas the conformation of the followee-ship reflects one’s personal interest and social intendment. It is intuitive that the follower-ship and followee-ship collaboratively define one’s unique social figure in virtual networks.

Besides, most of the existing work considers only local structural properties of the social networks for user alignment. Highly-order structural properties like the community structure are often ignored. We argue that how the anchor users involve in the communities of different networks can provide important hints to guide the alignment. The key idea is that it is not too hard for two different individuals in two different social networks to have common anchor users who are in the same community. However, if there are two individuals in two different social networks having common anchor users from different communities, it is more likely that they are the same person. Figure 1 gives an illustrated example. In the figure, network  $X$  and network  $Y$  have three “aligned” users/nodes (*Michael*, *Bob* and *Eric*), that is the anchor users. The friendship relations are indicated using arrowed lines. *Carl* in network  $X$  and *Bruce* in network  $Y$  share two anchor users: *Michael* and *Bob*. Meanwhile, *Carl* in network  $X$  and *Dr\_Carl* in network  $Y$  also share two anchor users: *Michael* and *Eric*. Suppose both *Michael* and *Bob* are from community 1 and *Eric* is from community 2. We consider that it is more likely for *Carl* and *Bruce* to share anchor users just by chance as both anchor users are in

• Li Liu, Xin Li and Lejian Liao are with Beijing Institute of Technology, Beijing, China, 100081. (The work is mostly done during Li Liu’s Ph.D. study at Beijing Institute of Technology and his visit to Hong Kong Baptist University.)

Corresponding author: Xin Li. E-mail: [xinli@bit.edu.cn](mailto:xinli@bit.edu.cn)

• William K. Cheung is with Hong Kong Baptist University.

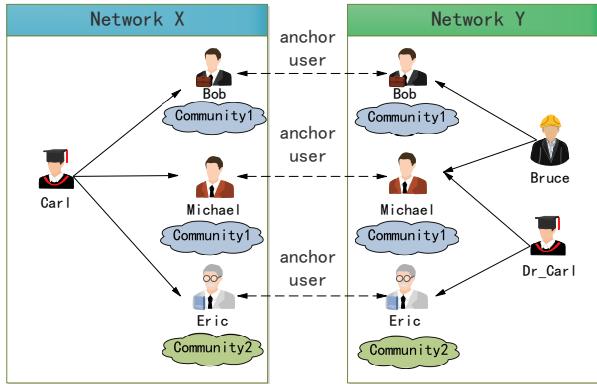


Fig. 1. A toy example to illustrate the “structural diversity” concept.

community 1. Instead, there will be a higher chance that *Carl* and *Dr\_Carl* are the same user as their two anchors belong to communities 1 and 2 respectively. We call this property for a set of sharing anchors to be its *structure diversity*. Referring to the same example in Figure 1, we consider the set of sharing anchors of  $\langle \text{Carl}, \text{Dr\_Carl} \rangle$  to have a higher structural diversity as that of  $\langle \text{Carl}, \text{Bruce} \rangle$ , and thus more likely to have the same identity. To further verify this intuition, we performed an empirical study on two real-world datasets Foursquare-twitter and DBLP (Refer to Table 2 for dataset description). We first detect communities [14] in two datasets. Then, based on given (true) anchors, we compute the average number of communities in which each anchor is involved (i.e., structural diversity). Also, we randomly sampled some false anchors and compute their structural diversity. According to Fig. 2 we observe that the former one (red dotted line) obviously has a much higher structural diversity value than the latter (blue dotted line).

Third, other than the modeling aspect, most of the existing work performs the network alignment task by using matrix factorization where matrix inverse operations or eigenvectors are typically involved, making them hard to scale up for large-scale problems. For instance, Wu *et. al* [12] presented the Consistent Incidence Co-Factorization (CICF) model for finding the pairs of latent factors in respective networks where a Projective Non-negative Matrix Factorization (PNMF) algorithm is utilized. Tan *et. al.* [13] proposed to construct hypergraphs for the social networks, followed by a manifold alignment step on these hypergraphs in a common embedding space. Computing the alignment is then equivalent to finding the second eigenvector of a specific matrix, which again is time-consuming for large graphs. Thus, a more scalable solution for aligning users across social networks is needed.

In this paper, we propose to extend the network representation learning (NLR) approach to social network alignment using a unified optimization framework where network embeddings of multiple social networks are learned simultaneously subject to some hard and soft constraints on common users among the networks. To contrast with some existing network representation learning methods [15], [16], the proposed network embedding explicitly models the follower-ship and followee-ship of each user as input and output context vector representations. We call it the *Input-*

*Output Network Embedding (IONE)*. The *IONE* can preserve the proximity of users with “similar” set of followers and followees in the embedding space. Also, both known and predicted user anchors across the networks can be introduced in a unified manner effectively as hard and soft constraints to facilitate the transfer of the contextual information across the networks. All the considering factors are formulated into a single objective function so that the network embedding and the user alignment can be jointly learned. We adopt negative sampling to address the scalability issue. Furthermore, we conducted a comprehensive evaluation to compare performance of a number of variants of *IONE* including *INE* (with only input context considered), *ONE* (with only output context considered), *IONE-D* (with structural diversity feature considered), *IONE-Con* (a concatenated version of *IONE* and *IONE-D*) etc., to analyze the importance of different factors to social network alignment. All our experiments are conducted based on real world datasets. The experimental results show that *IONE* framework consistently outperforms other competitive existing approaches. *IONE-Con* so far obtains the best performance among all *IONE*’s variants and further demonstrate the importance of properly modeling the follower/followee-ship and the structural diversity.

The main contributions of this paper include:

- We develop a unified framework named *IONE* to learn network representation for social network alignment where proximity relations of the users within individual networks and across them are preserved. Our proposed approach is capable of performing the feature representation learning and alignment simultaneously.
- We evaluate different variants of *IONE* where a number of design factors are being incorporated with the objective to enhance accuracy of the alignment task.
- We evaluate *IONE* and its variants with detailed experiments based on real world data. The results we obtained demonstrate significant improvement on user alignment accuracy in comparison with other state-of-the-art approaches.

## 2 RELATED WORK

Aligning two networks/graphs is not a new problem, and has found applications in different areas. For instance, related alignment tasks have been studied in bioinformatics for aligning protein-protein interaction networks [17]. Recent structural alignment methodologies proposed for social network alignment can roughly be categorized into *supervised* and *unsupervised*.

The unsupervised methods were proposed with the assumption of absence of anchor links, making the problem essentially a generic graph alignment problem. Koutra *et. al.* [18] focused on aligning *bipartite* graphs. They proposed an Alternating Projected Gradient Descent (APGD) based algorithm (called BIG-GLIGN), showing good performance on the MovieLens network alignment. Zhang *et. al.* [19] proposed a two-step algorithm where the anchor links are first inferred by *transitivity law* and *one-to-one* property, and then the anonymized networks are matched by minimizing the *friendship inconsistency*. In addition to the graph matching

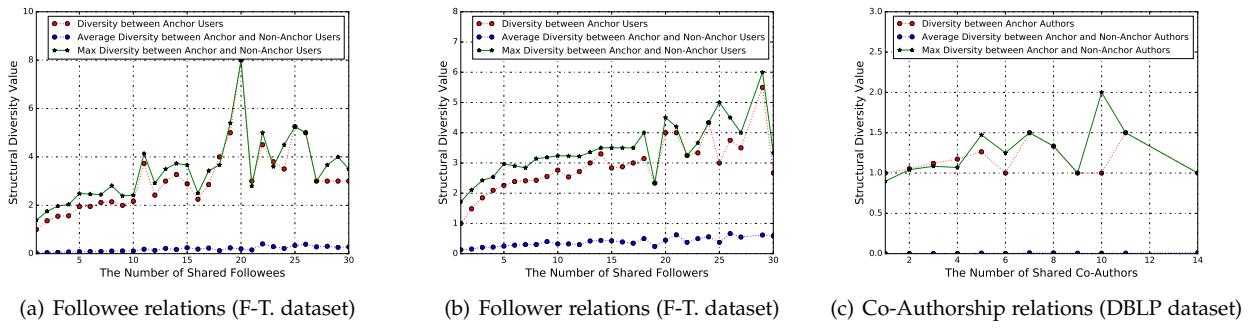


Fig. 2. Comparison of structural diversity values for true and false anchors

methods, some also make use of user profiles and structural features (i.e. username, locations, mutual friends) for unsupervised alignment. And these features are used to compute an affinity score matrix to obtain the alignment [7], [20], [21]. Representation learning methods were also proposed for unsupervised user alignment [22], [23], [24], in which the user embeddings are learned by utilizing users' relationships and attributes. Then methods like co-training [24], iterative process [22], [24] and xNetMF model [23] can be used for identifying users across networks.

The supervised learning methods for social network alignment fall into the following sub-categories: aggregating methods, classification methods, and embedding methods. Aggregating methods is meant to combine the similarity scores of different features (such as profile, content, and structural features) into a hybrid weight form function [25], [26], [27]. Among them, Nie *et al.* considered the dynamic change of people's core interests to define the structural feature and content feature, and obtained a unified similarity measure for aligning users [25].

There is another line of research which tries to predict the anchor links by using classification methods according to the feature vectors characterizing local topological properties and node attributes, such as username, description and profile images [8], [28], [29], [30], [31]. Zafarani *et al.* [10] explored the minimum feature of profiles (username). They extracted features based on human limitations, exogenous factors and endogenous factors, and then several popular classifiers were trained and validated for the best classifier selection. Kong *et al.* [32] first extended the structural features commonly used in link prediction, e.g., "common neighbors", "Jaccard's coefficient", and then trained an SVM classifier using the structural and heterogeneous features (e.g., spatial distribution, temporal distribution, and text content). By assuming one-to-one relationships between anchors, an MNA (Multi-Network Anchoring) model is derived to infer anchor links.

Learning network embeddings for supervised user alignment has been found quite promising [12], [33], [34]. The idea is to find a unified latent factor space for explaining the consistent behavior or structural features by utilizing the anchor seeds, which can better capture the heterogeneity across networks and avoid the tedious tasks of manually defining features. In [13], hyper-edges are firstly pre-defined. Then, the manifolds of two networks are projected onto the same embedding space in such a way that the

nodes within each hyper-edge are drawn together as close as possible in the embedding space, and the known aligned nodes (training data) should coincide in the projected space (as hard constraints). Eigenvalue decomposition on a matrix defined based on the hyperedges is required, which however is computational expensive. UMAH [35] is another model which represents social structures and user profile relations as a hypergraph for unified feature learning. Man *et. al* proposed PALE to learn embeddings of two networks individually while preserving the first-order proximity. Then, a supervised latent space matching is obtained by an MLP (Multi-Layer Perceptron) [36]. Zhang *et al.* [37] proposed a graph neural network model for alignment, in which the attribute embedding and structural embedding are incorporated into a convolutional neural network. Then a classifier is learned for predicting the alignment label.

Our work is similar in spirit to the methods which project multiple social networks into a common embedding space. To contrast, our proposed network embedding framework (to be detailed in the following sections) allows more social structural properties to be exploited for the representation learning, with the ultimate goal to result in more accurate and robust social network alignment results. In particular, the strength of the proposed framework lies in the incorporation of the explicit input and output context representations of nodes in the network to improve the expressive power of the network embedding model. Also, the users across the networks are projected onto a common embedding space during the representation learning progress without the need to introduce any latent space matching procedure. Moreover, high-order network features (such as structural diversity) can be easily incorporated into this embedding framework to further enhance its performance.

### 3 MODEL FRAMEWORK

Let  $G = (V, E, w)$  be a social network where  $V := \{v_i\}$  is the set of nodes representing the users and  $E := \{(v_i, v_j)\}$  is the set of directed edges representing their social relationships, and each edge is associated with a weight  $w_{ij} > 0$  indicating the tie strength.

#### 3.1 Input-Output Vector Representation

We propose a novel network embedding for representing social networks. Similar to most existing representation

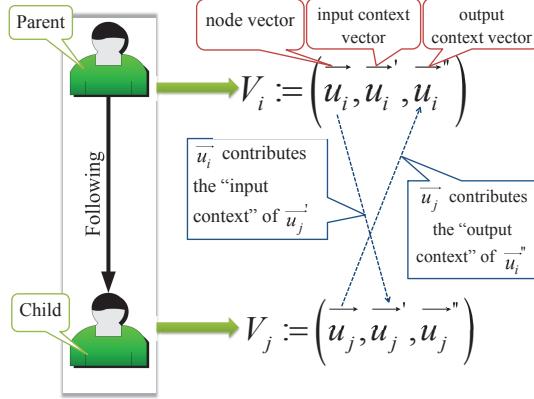


Fig. 3. An illustrated example of user representations

learning methods, we represent each node  $v_i \in V$  as a  $d$ -dimensional vector in an embedding space via a projection function  $f : V \rightarrow \mathbb{R}^d$ . By leveraging the follower-follower relations of the users, we allow the context of each user in a social network to be characterized by its set of followers and followees. To phrase that in a general network setting, we consider (i) the parents of a node as its *input context*, and at the same time (ii) its children as its *output context*. Accordingly, we represent each node  $v_i$  with three vector representations: a node vector  $\vec{u}_i \in \mathbb{R}^d$ , an input context vector  $\vec{u}'_i \in \mathbb{R}^d$ , and an output context vector  $\vec{u}''_i \in \mathbb{R}^d$ . As illustrated in Fig. 3,  $v_i$  is the parent node of  $v_j$  and thus  $\vec{u}_i$  contributes to  $\vec{u}'_j$  which represents the input context vector of  $v_j$ . Meanwhile, as  $v_j$  is at the same time the child of  $v_i$ ,  $\vec{u}_j$  contributes to  $\vec{u}''_i$  which represents the output context of  $v_i$ . As both input and output contexts are explicitly modelled for each node, we call the proposed model *Input-Output Network Embedding (IONE)*.

In order to learn the network embedding of a social network  $G$ , for each edge  $(v_i, v_j) \in E$ , we define the probability that  $v_i$  contributes specifically to  $v_j$  as its *input context* when compared with how  $v_i$  contributes to its other parent nodes, given as

$$p_1(v_j|v_i) = \frac{\exp(\vec{u}'_j \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{u}'_k \cdot \vec{u}_i)} \quad (1)$$

where  $|V|$  is the number of users in one network.

Similarly, we can define the probability that  $v_j$  contributes specially to  $v_i$  as its *output context* when compared with how  $v_j$  contributes to its other child nodes, given as

$$p_2(v_i|v_j) = \frac{\exp(\vec{u}''_i \cdot \vec{u}_j)}{\sum_{k=1}^{|V|} \exp(\vec{u}''_k \cdot \vec{u}_j)} \quad (2)$$

We further define the empirical counterparts of  $p_1(v_j|v_i)$  and  $p_2(v_i|v_j)$  as  $\hat{p}_1(v_j|v_i) = w_{ij}/d_i^{out}$  and  $\hat{p}_2(v_i|v_j) = w_{ij}/d_j^{in}$  respectively, where  $d_i^{out} = \sum_{k \in N_{out}(v_i)} w_{ik}$  and  $d_j^{in} = \sum_{k \in N_{in}(v_j)} w_{kj}$ . By minimizing the KL divergence of  $p_1$  and  $p_2$  and their empirical counterparts  $\hat{p}_1$  and  $\hat{p}_2$ , the corresponding IONE model can be inferred (as further detailed in the next section).

### 3.2 IONE: Input-Output Network Embedding

Given two social networks  $X$  and  $Y$ , we propose to compute their IONE models and at the same time align them. To do so, we first define an anchor node in one network (say  $X$ ) as a node which has a unique correspondence to another node in the other network (say  $Y$ ). The correspondence indicates that the two nodes are referring to the same person. We call the correspondence as an anchor link. Assuming that a set of anchor links bridging the two networks are available, we learn an aligned network embedding so that

- **Objective 1:** The structural proximity of the nodes within the two individual networks are preserved in their corresponding embeddings as far as possible, and
- **Objective 2:** The representations of the anchor nodes (i.e., nodes of the same user in the two networks) coincide in the embedding space, and those who are close in the embedded space can be considered as good candidates for user alignment.

Note that while our discussion assumes only two social networks (mainly for the clarity reason), the proposed method can naturally be extended to multiple networks.

To formulate the first objective, we minimize the KL-divergence of  $p_1$  and  $p_2$  and their empirical counterparts  $\hat{p}_1$  and  $\hat{p}_2$  over all the nodes in the two networks. By further defining the importance weighting of a parent node contributing to  $v_j$  as its input in network  $X$  as  $\lambda_{v_j}^{in}$  and the importance weighting of a child node contributing to  $v_i$  as its output in network  $X$  as  $\lambda_{v_i}^{out}$ , the corresponding objective function can be given as:

$$\begin{aligned} O_1 = & \sum_{v_j \in V^{X/Y}} \lambda_{v_j}^{out} KL(\hat{p}_1^{X/Y}(v_j|v_i) || p_1(v_j^{X/Y}|v_i^{X/Y})) \\ & + \sum_{v_i \in V^{X/Y}} \lambda_{v_i}^{in} KL(\hat{p}_2^{X/Y}(v_i|v_j) || p_2(v_i^{X/Y}|v_j^{X/Y})) \end{aligned} \quad (3)$$

By setting  $\lambda_i^{out}$  as the output-degree  $d_i^{out}$  of  $v_i$  and  $\lambda_j^{in}$  as the input-degree  $d_j^{in}$  of  $v_j$ , the objective function is rewritten as:

$$\begin{aligned} O_1 = & - \sum_{(v_i, v_j) \in E^{X/Y}} w_{ij}^{X/Y} \log p_1(v_j^{X/Y}|v_i^{X/Y}) \\ & - \sum_{(v_i, v_j) \in E^{X/Y}} w_{ij}^{X/Y} \log p_2(v_i^{X/Y}|v_j^{X/Y}) \end{aligned} \quad (4)$$

To formulate the second objective, we set the node vector representations of the corresponding anchor nodes in the two networks to be identical as hard constraints. In other words, we make the IONE embeddings of the  $X$  and  $Y$  aligned at the anchor nodes. On top of that, to enhance alignment accuracy for nodes other than the anchor ones, we train a classifier for anchor link prediction so that the anchor link prediction results can be used as “soft” constraints. The structural features used for the classifier training include common neighbors, extended Jaccard’s coefficient, and extended Adamic/Adar measures. To implement both, we adopt a second objective function.

We denote  $p_a(v_i^X|v_k^Y)$  as the probability that  $v_i^X$  and  $v_k^Y$  are the same user as predicted by a pre-trained classifier. If say there is an anchor link between  $v_i^X$  and  $v_k^Y$  as provided

in the training set, we set the value of the corresponding  $p_a$  to 1. It can be proved that this is equivalent to setting the hard constraints for the representations of the corresponding anchor users to be identical (See Appendix A.). For non-anchor nodes, the estimated  $p_a(v_i^X|v_k^Y)$  acts as a “bridge” between  $v_i^X$  and  $v_k^Y$ , and  $v_i^X$  in network  $X$  can contribute as its input context to the  $v_k^Y$ ’s children in network  $Y$  with the probability  $p_a(v_i^X|v_k^Y)$ . Similarly,  $v_k^Y$  in network  $Y$  can contribute to  $v_i^X$ ’s children in network  $X$  as if it is  $v_i^X$  with a probability  $p_a(v_i^X|v_k^Y)$ . Based on this idea, we first define their empirical probabilities:

$$\hat{p}_1(v_j^Y|v_i^X) = \sum_{(v_k, v_j) \in E^Y} w_{kj}^Y * p_a(v_i^X|v_k^Y) / d_k^{outY} \quad (5)$$

$$\hat{p}_2(v_i^X|v_j^Y) = \sum_{(v_k, v_j) \in E^Y} w_{kj}^Y * p_a(v_i^X|v_k^Y) / d_j^{inY} \quad (6)$$

$$\hat{p}_1(v_j^Y|v_i^X) = \sum_{(v_k, v_j) \in E^X} w_{kj}^X * p_a(v_i^Y|v_k^X) / d_k^{outX} \quad (7)$$

$$\hat{p}_2(v_i^Y|v_j^X) = \sum_{(v_k, v_j) \in E^X} w_{kj}^X * p_a(v_i^Y|v_k^X) / d_j^{inX}. \quad (8)$$

Again, we minimize the KL-divergence of  $p$  and the empirical distribution  $\hat{p}$ . Similar to the objective function  $O_1$  with the output-degree  $d_k^{outY}$  of  $v_k^Y$  and the input-degree  $d_j^{inY}$  of  $v_j^Y$  used as the importance weightings, the objective function can be rewritten as:

$$\begin{aligned} O_2 = & - \sum_{v_i \in X} \sum_{(v_k, v_j) \in E^Y} w_{kj}^Y p_a(v_i^X|v_k^Y) \log p_1(v_j^Y|v_i^X) \\ & - \sum_{v_i \in X} \sum_{(v_k, v_j) \in E^Y} w_{kj}^Y p_a(v_i^X|v_k^Y) \log p_2(v_i^X|v_j^Y) \\ & - \sum_{v_i \in Y} \sum_{(v_k, v_j) \in E^X} w_{kj}^X p_a(v_i^Y|v_k^X) \log p_1(v_j^X|v_i^Y) \\ & - \sum_{v_i \in Y} \sum_{(v_k, v_j) \in E^X} w_{kj}^X p_a(v_i^Y|v_k^X) \log p_2(v_i^Y|v_j^X). \end{aligned} \quad (9)$$

Thus, the IONE framework with the soft constraints (or hard constraints if  $p_a$  set to 1) of the multiple networks in the proposed network embedding space can be computed by minimizing a combined objective function  $O = O_1 + O_2$  w.r.t.  $\{\overrightarrow{u_x^X}, \overrightarrow{u_x^Y}, \overrightarrow{u_x^Z}, \overrightarrow{u_y^Y}, \overrightarrow{u_y^Z}, \overrightarrow{u_y^X}\}$  where  $O_1$  helps ensure that two nodes (users) sharing more common input and output contexts (common followers and followees) will be drawn closer in the embedded space, and  $O_2$  allows the contexts of the anchor nodes to be propagated across the networks which in turn can enhance the learning of the embeddings of the two individual networks. In the sequel, we called IONE with the classifier output as the soft constraints as IONE-S and the version without the soft constraints simply IONE.

### 3.3 IONE-D: Input-Output Network Embedding with Structural Diversity Incorporated

IONE utilizes the second-order proximity [16] to drive nodes with more mutual neighbors closer in the low-dimensional embedding space for detecting the potential candidates of anchor users. However, it is also not too hard for two distinct users in different networks to know the same set

of anchor users if the anchor users come from the same community. But if the set of anchor users are coming from different communities, the chance that the distinct users are in fact the same person will be higher. Thus, simply adopting the original form of second-order proximity may be not good enough, which triggers us to include more informative cues. We propose to explore the community structure of the social networks and incorporate a notion of structural diversity into the IONE framework (referred to as IONE-D) for characterizing a set of anchor users.

We first apply an existing community detection algorithm [14] to one of the networks. Then, each node (including the anchor nodes) will be assigned to a particular community. We then define a diversity value  $D_{v_i^X|v_a|v_j^Y}$  for each anchor  $v_a \in S_{anchor}^{i,j}$  where  $S_{anchor}^{i,j}$  is the set of anchor users whose elements are shared by  $v_i^X$  and  $v_j^Y$ ,  $v_i^X$  and  $v_j^Y$  are the non-anchor nodes in networks  $X$  and  $Y$ . For networks with directed edges, according to the role of the anchor being a follower or a followee, the anchor set can be defined as  $S_{anchor}^{i,j,follower}$  and  $S_{anchor}^{i,j,followee}$ . Two different diversity measures are defined, given as

$$D_{v_i^X|v_a|v_j^Y}^{follower} = NDC(S_{anchor}^{i,j,follower}) \quad (10)$$

for  $v_a \in S_{anchor}^{i,j,follower}$ ,  $v_a \rightarrow v_i^X$  and  $v_a \rightarrow v_j^Y$ , and

$$D_{v_i^X|v_a|v_j^Y}^{followee} = NDC(S_{anchor}^{i,j,followee}) \quad (11)$$

for  $v_a \in S_{anchor}^{i,j,followee}$ ,  $v_a \leftarrow v_i^X$  and  $v_a \leftarrow v_j^Y$ .  $NDC(S_{anchor}^{i,j,follower/followee})$  refers to the number of distinct communities which the shared anchors belong. We then incorporate the structural diversity measures and introduce IONE-D with one more objective being considered:

- Objective 3:** The nodes with “similar” sets of anchor nodes are close in the embedding space, and the anchor nodes with higher structural diversity values will contribute more to hint the alignment of a pair of users who share them.

To formulate this objective, we essentially multiply the edge weight with the diversity value where an anchor node is involved. So, for each across-network user pair  $(v_i^X, v_j^Y)$  and their shared anchor  $v_a$ , we define the following empirical probabilities:

$$\begin{aligned} \hat{p}_1(v_a|v_i^X) &= w_{ia} * D_{v_i^X|v_a|v_j^Y}^{followee} / Z_i^{out} \\ \hat{p}_1(v_i^X|v_a) &= w_{ai} * D_{v_i^X|v_a|v_j^Y}^{follower} / Z_a^{outX} \\ \hat{p}_1(v_a|v_j^Y) &= w_{ja} * D_{v_i^X|v_a|v_j^Y}^{followee} / Z_j^{out} \\ \hat{p}_1(v_j^Y|v_a) &= w_{aj} * D_{v_i^X|v_a|v_j^Y}^{follower} / Z_a^{outY} \\ \hat{p}_2(v_i^X|v_a) &= w_{ia} * D_{v_i^X|v_a|v_j^Y}^{followee} / Z_a^{inX} \\ \hat{p}_2(v_a|v_i^X) &= w_{ai} * D_{v_i^X|v_a|v_j^Y}^{follower} / Z_i^{in} \\ \hat{p}_2(v_j^Y|v_a) &= w_{ja} * D_{v_i^X|v_a|v_j^Y}^{followee} / Z_a^{inY} \\ \hat{p}_2(v_a|v_j^Y) &= w_{aj} * D_{v_i^X|v_a|v_j^Y}^{follower} / Z_j^{in} \end{aligned} \quad (12)$$

where  $Z_*$ s are the normalized factors. By minimizing the KL divergence of  $p_1$  and  $p_2$  and their empirical counterparts  $\hat{p}_1$

and  $\hat{p}_2$  in networks  $X$  and  $Y$  respectively, the corresponding objective function is given as:

$$O_3 = \sum_{v_a \in \text{anchor}} \sum_{(v_i, v_a) \in E^X} \sum_{(v_j, v_a) \in E^Y} \left\{ \begin{array}{l} \lambda_i^{out} KL(\hat{p}_1(v_a|v_i^X) || p_1(v_a|v_i^X)) \\ + \lambda_j^{out} KL(\hat{p}_1(v_a|v_j^Y) || p_1(v_a|v_j^Y)) \\ + \lambda_a^{inX} KL(\hat{p}_2(v_i^X|v_a) || p_2(v_i^X|v_a)) \\ + \lambda_a^{inY} KL(\hat{p}_2(v_j^Y|v_a) || p_2(v_j^Y|v_a)) \end{array} \right. \\ \left. + \sum_{v_a \in \text{anchor}} \sum_{(v_a, v_i) \in E^X} \sum_{(v_a, v_j) \in E^Y} \left\{ \begin{array}{l} \lambda_a^{outX} KL(\hat{p}_1(v_i^X|v_a) || p_1(v_i^X|v_a)) \\ + \lambda_a^{outY} KL(\hat{p}_1(v_j^Y|v_a) || p_1(v_j^Y|v_a)) \\ + \lambda_i^{in} KL(\hat{p}_2(v_a|v_i^X) || p_2(v_a|v_i^X)) \\ + \lambda_j^{in} KL(\hat{p}_2(v_a|v_j^Y) || p_2(v_a|v_j^Y)) \end{array} \right. \right\} \quad (13)$$

By setting  $\lambda_i^{out}$  as  $Z_i^{out}$ ,  $\lambda_j^{out}$  as  $Z_j^{out}$ , et al., the objective function can be rewritten as:

$$O_3 = - \sum_{v_a \in \text{anchor}} \sum_{(v_i, v_a) \in E^X} \sum_{(v_j, v_a) \in E^Y} D_{v_i^X|v_a|v_j^Y} * \left\{ \begin{array}{l} w_{ia}^X \log p_1(v_a|v_i^X) + w_{ja}^Y \log p_1(v_a|v_j^Y) \\ w_{ia}^X \log p_2(v_i^X|v_a) + w_{ja}^Y \log p_2(v_j^Y|v_a) \end{array} \right. \\ - \sum_{v_a \in \text{anchor}} \sum_{(v_a, v_i) \in E^X} \sum_{(v_a, v_j) \in E^Y} D_{v_i^X|v_a|v_j^Y} * \left\{ \begin{array}{l} w_{ai}^X \log p_1(v_i^X|v_a) + w_{aj}^Y \log p_1(v_j^Y|v_a) \\ w_{ai}^X \log p_2(v_a|v_i^X) + w_{aj}^Y \log p_2(v_a|v_j^Y) \end{array} \right. \quad (14)$$

If one wants to combine the ideas behind *IONE-D* and *IONE-S*, one possibility is to start with *IONE-S* where the probability for each node to be linked with an anchor is first estimated as the soft constraints. Then, a structural diversity value defined based on those soft constraints will need to be defined. In this paper, we combine *IONE-S* and *IONE-D* by simply taking the structural diversity measure as an additional feature into the pre-trained classifier for obtaining the probability  $p_a(v_i^X|v_j^Y)$ . In particular, the structural features used for the classifier training include the number of common neighbors, the structural diversity values of neighbors, Adamic/Adar measures weighted by the diversity values. After that, we incorporate  $p_a$  into *IONE* framework as soft constraints again. We name this version of *IONE* as *IONE-D-S*.

### 3.4 Model Inference

We use the stochastic gradient descent to learn the vector representations of the two networks. Note that there is a difference between *IONE* and *IONE-D* for the representation learning. In *IONE*, only two nodes are involved in one updating step as the weight  $w_{ij}$  is determined by  $v_i$  and  $v_j$ . But for *IONE-D*, a triple with three nodes must be sampled in each updating step as the diversity value  $D_{v_i^X|v_a|v_j^Y}$  is determined by two across-network nodes  $v_i^X$  and  $v_j^Y$ , and an anchor node  $v_a$ . In the following two subsections, we will describe the model inference steps for learning the two models.

#### 3.4.1 Model Inference For *IONE*

For *IONE* framework, to update the vector representations of  $v_i$  and  $v_j$  in network  $X$ , i.e.,  $\vec{u}_i^X, \vec{u}_j^X$ , the gradient can be computed as:

$$\frac{\partial O}{\partial \vec{u}_i^X} = \frac{\partial O_1}{\partial \vec{u}_i^X} + \frac{\partial O_2}{\partial \vec{u}_i^X} \\ = \sum_{(v_i, v_j) \in E^X} w_{ij}^X * \frac{\partial \log p_1(v_j^X|v_i^X)}{\partial \vec{u}_i^X} \\ + \sum_{v_i \in V^X} \sum_{(v_k, v_j) \in E^Y} w_{kj}^Y * p_a(v_i^X|v_k^Y) \frac{\partial \log p_1(v_j^Y|v_i^X)}{\partial \vec{u}_i^X} \quad (15)$$

$$\frac{\partial O}{\partial \vec{u}_j^X} = \frac{\partial O_1}{\partial \vec{u}_j^X} + \frac{\partial O_2}{\partial \vec{u}_j^X} \\ = \sum_{(v_i, v_j) \in E^X} w_{ij}^X * \frac{\partial \log p_2(v_i^X|v_j^X)}{\partial \vec{u}_j^X} \\ + \sum_{v_i \in V^Y} \sum_{(v_k, v_j) \in E^X} w_{kj}^X * p_a(v_i^Y|v_k^X) \frac{\partial \log p_2(v_i^Y|v_j^X)}{\partial \vec{u}_j^X}. \quad (16)$$

The partial derivatives w.r.t. the input and output context vectors can be obtained similarly.

Calculating the conditional probabilities  $p_1$  and  $p_2$  in Eqs. (15) and (16) require the summation over the entire set of nodes. To reduce the computational complexity, we adopt the negative sampling approach [38] that makes the problem of minimizing the above objective function equivalent to a problem of estimating the parameters of a probabilistic binary classifier that uses the same parameters to distinguish samples of the empirical distribution from samples generated by the noise distribution. The equivalent counterparts of the objective function can be derived as:

$$\log p_1(v_j^X|v_i^X) \propto \log \sigma(\vec{u}_j'^{X^T} \cdot \vec{u}_i^X) \\ + \sum_{m=1}^K E_{v_n \sim p_n(v)} \log \sigma(-\vec{u}_n'^{X^T} \cdot \vec{u}_i^X) \quad (17)$$

$$\log p_1(v_j^Y|v_i^X) \propto \log \sigma(\vec{u}_j'^{Y^T} \cdot \vec{u}_i^X) \\ + \sum_{m=1}^K E_{v_n \sim p_n(v)} \log \sigma(-\vec{u}_n'^{Y^T} \cdot \vec{u}_i^X) \quad (18)$$

$$\log p_2(v_i^X|v_j^Y) \propto \log \sigma(\vec{u}_i'^{X^T} \cdot \vec{u}_j^X) \\ + \sum_{m=1}^K E_{v_n \sim p_n(v)} \log \sigma(-\vec{u}_n'^{X^T} \cdot \vec{u}_j^X) \quad (19)$$

$$\log p_2(v_i^Y|v_j^X) \propto \log \sigma(\vec{u}_i'^{Y^T} \cdot \vec{u}_j^X) \\ + \sum_{m=1}^K E_{v_n \sim p_n(v)} \log \sigma(-\vec{u}_n'^{Y^T} \cdot \vec{u}_j^X) \quad (20)$$

where  $\sigma(x) = 1/(1+\exp(-x))$  is the sigmoid function,  $K$  is the number of negative samples  $v_n$  which is sampled from the “noisy distribution” of  $p_n(v) = d_v^{3/4}$  as in [38], and  $d_v$  is the output degree.

Since there might be a high variance of edge weights  $w$ , it is hard to find a reasonable learning rate. In order to solve this problem, we unfold a weighted edge into multiple binary edges. Moreover the alias table [39] is utilized for

edge sampling. With the edge-sampling method adopted, the weight  $w_{ij}$  can be ignored when calculating Eqs.(15) and (16) as it is implicitly included in the sampling process.

With the negative sampling and the alias table adopted, once that  $v_i, v_j$  in one edge and a negative node  $v_n$  are sampled in two networks, the partial derivative of Eq.(15) w.r.t.  $\vec{u}_i^X$  can be rewritten as:

$$\begin{aligned} \frac{\partial O}{\partial \vec{u}_i^X} &= [1 - \sigma(\vec{u}_j'^X T \cdot \vec{u}_i^X)] \vec{u}_j'^X - \sigma(\vec{u}_n'^X T \cdot \vec{u}_i^X) \vec{u}_n'^X \\ &\quad + \sum_{(v_k, v_j) \in E^Y} p_a(v_i^X | v_k^Y) * \\ &\quad \{[1 - \sigma(\vec{u}_j'^Y T \cdot \vec{u}_i^X)] \vec{u}_j'^Y - \sigma(\vec{u}_n'^Y T \cdot \vec{u}_i^X) \vec{u}_n'^Y\} \end{aligned} \quad (21)$$

Similarly, we can obtain the partial derivatives w.r.t. the other context vectors of the concerned nodes given as:

$$\begin{aligned} \frac{\partial O}{\partial \vec{u}_j^X} &= [1 - \sigma(\vec{u}_i''^X T \cdot \vec{u}_j^X)] \vec{u}_i''^X - \sigma(\vec{u}_n''^X T \cdot \vec{u}_j^X) \vec{u}_n''^X \\ &\quad + \sum_{(v_k, v_j) \in E^X} p_a(v_i^Y | v_k^X) * \\ &\quad \{[1 - \sigma(\vec{u}_i''^Y T \cdot \vec{u}_k^X)] \vec{u}_i''^Y - \sigma(\vec{u}_n''^Y T \cdot \vec{u}_k^X) \vec{u}_n''^Y\} \end{aligned} \quad (22)$$

$$\begin{aligned} \frac{\partial O}{\partial \vec{u}_i''^X} &= [1 - \sigma(\vec{u}_i''^X T \cdot \vec{u}_j^X)] \vec{u}_j^X + \sum_{(v_k, v_j) \in E^Y} p_a(v_i^X | v_k^Y) \\ &\quad * [1 - \sigma(\vec{u}_i''^X T \cdot \vec{u}_j^Y)] \vec{u}_j^Y \end{aligned} \quad (23)$$

$$\begin{aligned} \frac{\partial O}{\partial \vec{u}_j''^X} &= [1 - \sigma(\vec{u}_j'^X T \cdot \vec{u}_i^X)] \vec{u}_i^X + \sum_{(v_k, v_j) \in E^X} p_a(v_i^Y | v_k^X) \\ &\quad * [1 - \sigma(\vec{u}_j'^X T \cdot \vec{u}_i^Y)] \vec{u}_i^Y \end{aligned} \quad (24)$$

$$\begin{aligned} \frac{\partial O}{\partial \vec{u}_n''^X} &= -\sigma(\vec{u}_n'^X T \cdot \vec{u}_i^X) \vec{u}_i^X + \sum_{(v_k, v_j) \in E^X} p_a(v_i^Y | v_k^X) \\ &\quad * [-\sigma(\vec{u}_n'^X T \cdot \vec{u}_i^Y) \vec{u}_i^Y] \end{aligned} \quad (25)$$

$$\begin{aligned} \frac{\partial O}{\partial \vec{u}_n''^X} &= -\sigma(\vec{u}_n''^X T \cdot \vec{u}_j^X) \vec{u}_j^X + \sum_{(v_k, v_j) \in E^Y} p_a(v_i^X | v_k^Y) \\ &\quad * [-\sigma(\vec{u}_n''^X T \cdot \vec{u}_j^Y) \vec{u}_j^Y] \end{aligned} \quad (26)$$

With reference to Eqs.(21-26), the updating rules for network  $Y$  can be obtained by swapping the superscripts  $X$  and  $Y$ . The overall learning algorithm for *IONE* is shown in Algorithm 1.

### 3.4.2 Model Inference For *IONE-D*

For *IONE-D*, the diversity value  $D_{v_i^X | v_a | v_j^Y}$  is defined based on three nodes  $v_i^X, v_j^Y$  and  $v_a$ . Thus, triple sampling is needed to be first performed as preprocessing. Specifically, we consider a cross-network path  $v_i^X, v_a, v_j^Y$  and set the weight of the path as  $(w_{ia} + w_{ja}) * D_{v_i^X | v_a | v_j^Y}$ . Then we sample triple instances from the alias table which is now constructed by the cross network path. Again the stochastic gradient descent is used for parameter learning.

Similar to the learning of *IONE*, negative sampling is utilized. Note that the main difference between *IONE* and

---

### Algorithm 1 Learning Aligned Input-Output Network Embeddings of Two Networks (*IONE*)

---

**Require:** Two networks  $G^X$  and  $G^Y$ , a set of anchor links  $E_a$ , learning rate  $\eta$ , # of negative samples  $K$

**Ensure:** The set of estimated parameters  $\Theta = \{\vec{u}_i^X, \vec{u}_j^X, \vec{u}_i''^X, \vec{u}_j''^X, \vec{u}_n^X, \vec{u}_n''^X, \vec{u}_i^Y, \vec{u}_j^Y, \vec{u}_i''^Y, \vec{u}_j''^Y, \vec{u}_n^Y, \vec{u}_n''^Y\}$

```

1: procedure LEARNING( $G^X, G^Y, E_a, \eta, K$ )
2:   Initialize  $\Theta = \{\vec{u}_x^X, \vec{u}_x'^X, \vec{u}_x''^X, \vec{u}_y^Y, \vec{u}_y'^Y, \vec{u}_y''^Y\}$ 
3:   repeat
4:     for  $N$  in  $(X, Y)$  do
5:       Sample one edge  $(v_i, v_j)$  from  $G^N$ 
6:       Update  $\vec{u}_i^X, \vec{u}_j^X, \vec{u}_i''^X, \vec{u}_j''^X$  based on Eqs.(21-24)
7:       for  $i = 0; i < K; i = i + 1$  do
8:         Sample a negative node  $v_n$ 
9:         Update  $\vec{u}_i^X, \vec{u}_j^X, \vec{u}_i''^X, \vec{u}_j''^X$  based on Eqs.(21,
22, 25, 26)
10:        end for
11:      end for
12:    until convergence
13:    return  $\Theta$ 
14: end procedure

```

---

*IONE-D* is the adoption of the diversity value for the path  $(v_a, v_i^X, v_j^Y)$ . Thus we need to sample edges which construct such paths. The gradients are then rewritten as:

$$\frac{\partial O_3}{\partial \vec{u}_i^X} = [1 - \sigma(\vec{u}_a'^T \cdot \vec{u}_i^X)] \vec{u}_a' - \sigma(\vec{u}_n'^X T \cdot \vec{u}_i^X) \vec{u}_n'^X + [1 - \sigma(\vec{u}_a''^T \cdot \vec{u}_i^X)] \vec{u}_a'' - \sigma(\vec{u}_n''^X T \cdot \vec{u}_i^X) \vec{u}_n'' \quad (27)$$

$$\frac{\partial O_3}{\partial \vec{u}_j^Y} = [1 - \sigma(\vec{u}_a'^T \cdot \vec{u}_j^Y)] \vec{u}_a' - \sigma(\vec{u}_n'^Y T \cdot \vec{u}_j^Y) \vec{u}_n'^Y + [1 - \sigma(\vec{u}_a''^T \cdot \vec{u}_j^Y)] \vec{u}_a'' - \sigma(\vec{u}_n''^Y T \cdot \vec{u}_j^Y) \vec{u}_n'' \quad (28)$$

$$\begin{aligned} \frac{\partial O_3}{\partial \vec{u}_a} &= [1 - \sigma(\vec{u}_i''^X T \cdot \vec{u}_a)] \vec{u}_i''^X - \sigma(\vec{u}_n''^X T \cdot \vec{u}_a) \vec{u}_n''^X + [1 - \sigma(\vec{u}_j''^Y T \cdot \vec{u}_a)] \vec{u}_j''^Y - \sigma(\vec{u}_n''^Y T \cdot \vec{u}_a) \vec{u}_n''^Y + [1 - \sigma(\vec{u}_i'^X T \cdot \vec{u}_a)] \vec{u}_i'^X - \sigma(\vec{u}_n'^X T \cdot \vec{u}_a) \vec{u}_n'^X + [1 - \sigma(\vec{u}_j'^Y T \cdot \vec{u}_a)] \vec{u}_j'^Y - \sigma(\vec{u}_n'^Y T \cdot \vec{u}_a) \vec{u}_n'^Y \end{aligned} \quad (29)$$

The gradient of  $O_3$  w.r.t other parameters in the form of negative sampling can be obtained similarly. The overall learning algorithm for *IONE-D* is shown in Algorithm 2.

### 3.5 Time Complexity

Here, we conduct a theoretical analysis on computational complexity to justify the efficiency of our proposed methods. Sampling an edge takes constant time  $O(1)$ . Optimization using negative sampling takes  $O(d(K+1))$  time, where  $K$  is the number of negative samples. Therefore, the overall complexity for each step is  $O(dK)$ . In practice, the number of steps need for the optimization is proportional to the number of edges  $O(|E|)$ . Therefore, the overall time complexity of our model is  $O(dK|E|)$  which is linear to the number of edges  $|E|$  and does not depend on the number of nodes  $|V|$ .

---

**Algorithm 2** Learning Aligned Input-Output Network Embeddings With Community Diversity (IONE-D)

---

**Require:** Two networks  $G^X$  and  $G^Y$ , a set of anchor links  $E_a$ , learning rate  $\eta$ , # of negative samples  $K$

**Ensure:** The set of estimated parameters  $\Theta = \{\vec{u}_i, \vec{u}_j, \vec{u}_i'', \vec{u}_j'', \vec{u}_n', \vec{u}_n''\}$

- 1: **procedure** LEARNING( $G^X, G^Y, E_a, \eta, K$ )
- 2:    Re-form all the edges based on anchor links  $E_a$ , the edges type are  $v_i^X \rightarrow v_a \leftarrow v_j^Y$  and  $v_i^X \leftarrow v_a \rightarrow v_j^Y$  based on the direction of relationship between users and the anchor.
- 3:    Initialize  $\Theta = \{\vec{u}_x^X, \vec{u}_x'^X, \vec{u}_x''^X, \vec{u}_y^Y, \vec{u}_y'^Y, \vec{u}_y''^Y\}$
- 4:    **repeat**
- 5:     Sample one re-formed edge
- 6:     **for**  $N$  in  $(X, Y)$  **do**
- 7:       Update  $\vec{u}_i, \vec{u}_j, \vec{u}_a$  based on Eqs.(27-29). Note that, if the re-formed edge type accords with the aforementioned first case, using the first-half part of Eqs.(27-29). Similarly, the second case uses the second-half part.
- 8:       **for**  $i = 0; i < K; i = i + 1$  **do**
- 9:         Sample a negative node  $v_n$
- 10:         Update the rest parameters similarly with Eqs.(27-29).
- 11:       **end for**
- 12:     **end for**
- 13:     **until** convergence
- 14:     **return**  $\Theta$
- 15: **end procedure**

---

### 3.6 Mapping Users Across Social Networks

To map users across different social networks, we compute the cosine similarity between the vector representations of one node in network  $X$  and another in network  $Y$  to determine the correspondence.

$$rel(v_i^X, v_j^Y) = \frac{\sum_{p=1}^d u_{ip}^X \times u_{jp}^Y}{\sqrt{\sum_{p=1}^d u_{ip}^{X^2}} \times \sqrt{\sum_{p=1}^d u_{jp}^{Y^2}}} \quad (30)$$

So, for each user  $v_i^X$  in network  $X$ , we can determine whether  $v_j^Y$  in network  $Y$  is a potential anchor candidate based on the ranking of the defined relevance.

## 4 EXPERIMENTS

For performance evaluation, we employ two real-world cross network datasets collected from Foursquare, Twitter [11], [20], [32] (T-F), and DBLP<sup>1</sup> (DBLP). We obtain the ground truth of twitter and foursquare anchor users in the T-F dataset by noting that some Foursquare users provide their twitter accounts in their profiles. For the DBLP dataset, authors are split into different co-author networks by filtering publication venues of their papers. The first network contains authors who published papers in "Data Mining" related conferences or journals including WWW, SIGKDD, PAKDD, TKDD and etc. The other one consists of authors who published papers in "Machine Learning" related venues such as NIPS, ICML, ICONIP, ICMLA, ICMLC

and etc. Note that the co-author relationships are non-directional, we transform them by making co-authors as followers/followees of each other. The ground truth anchors of this dataset are the authors who published papers in the both areas. Table 2 lists out the statistics of these datasets.

### 4.1 Comparative Methods

We compare the proposed models with several state-of-the-art methods, which are summarized as follows:

- **MAG**: A graph-based manifold alignment [13] where the similarity of a linked user pair is defined as 1.
- **MAH**: A hypergraph-based manifold alignment [13] where the hyperedges model the high-order relations in a social network.
- **CLF**: A method called collective link fusion [11] which includes two phases: 1) collective multi-network link prediction and 2) collective link fusion across partially aligned probabilistic networks.
- **CRW**: A method called collective random walk with restart that is essentially the second step of **CLF**, which is used here as a baseline.
- **PALE\_LINE**: An embedding based approach [36] where the embeddings of individual networks are learned using LINE [16], and an MLP is used for projecting the embeddings to the target space.
- **PALE\_DeepWalk**: A variant of embedding based model proposed in [36], in which DeepWalk [15] is adopted for learning individual network embeddings.
- **ULINK**: A model for the multi-platform user identity linkage based on Latent User Space [33], in which a constrained concave-convex procedure is adopted for the optimization.
- **FRUI**: A Friend Relationship-Based User Identification algorithm across networks [40]. For directed social networks, we use the union set of followers and followees as friend relationships.
- **BootEA**: A bootstrapping approach for embedding-based entity alignment [34], which iteratively labels potential aligned entities as training data for learning alignment-oriented KG embeddings.

We name our proposed model with only hard constraints on anchor nodes as *IONE* and that with also the soft constraints as explained in Sec.3.2 as *IONE-S*. We name the proposed model which uses input context and node vectors as *INE* and name the model that uses output context and node vectors as *ONE*. For the IONE framework with community diversity considered, we name it as *IONE-D*. Then, a combined model which simply concatenates *IONE* embedding and *IONE-D* embedding is abbreviated as *IONE-Con*. Moreover, we also investigate the importance of considering also the context vectors inferred and extend the current representation of nodes by concatenating their input and output context representations to result in *IONE-Ex*, *IONE-D-Ex* and *IONE-Con-Ex*. Table 1 tabulates the representations and extra information being adopted in the different versions of IONE which are to be evaluated in this paper. For the community detection, we use an open source api.<sup>2</sup> "best\_partition" function with all parameters

1. <https://www.aminer.cn/citation>

2. <https://perso.crans.org/aynaud/communities/api.html>

TABLE 1  
Usage of Embeddings and Information For All Proposed Models

Embeddings or Information \ Models	IONE	IONE-S	IONE-Ex	IONE-D	IONE-D-S	IONE-D-Ex	IONE-Con	IONE-Con-Ex
Node Vector (learned by IONE)	✓	✓	✓		✓		✓	✓
Input Context Vector (learned by IONE)			✓					✓
Output Context Vector (learned by IONE)			✓					✓
Soft Constraint		✓			✓			
Structure Diversity				✓	✓	✓	✓	✓
Node Vector (learned by IONE-D)				✓	✓	✓	✓	✓
Input Context Vector (learned by IONE-D)					✓			✓
Output Context Vector (learned by IONE-D)						✓		✓

TABLE 2  
Statistics of The Datasets Used for Evaluation

Networks	#Users	#Relations	#Anchors
Twitter	5,220	164,919	1,609
	5,315	76,972	
Foursquare	11,526	47,326	1,295
	12,311	43,948	
DBLP_DataMining			
DBLP_MachineLearning			

set as provided default value are adopted for heuristically determining the community. The learning rates of SGD are empirically set as 0.01 for IONE and 0.03 for IONE-D.<sup>3</sup>

## 4.2 Evaluation Metrics

In our experiments, *Precision@N* is the evaluation metric, given as:

$$\text{Precision}@N = \frac{|\text{CorrUser}@N|^X + |\text{CorrUser}@N|^Y}{|\text{UnMappedAnchors}| \times 2} \quad (31)$$

where  $|\text{CorrUser}@N|$  is the number of unmapped anchor users with their corresponding users found among the top- $N$  neighbors in the embedded space.  $|\text{UnMappedAnchors}|$  is the total number of all unmapped anchor users. Different from some baselines such as FRUI which tries to predict the labels of sampling pairs given a candidate anchor user, we do the evaluation by first ranking all the candidate pairs (both the positive and negative pairs which are related to the candidate anchor user).

Also, since the network alignment performance in general depends on the degree of overlapping of the two networks, we measure the degree as in [13], given as

$$\text{Interop}(X, Y) = \frac{|\text{Correlations}| \times 2}{|\text{Relations}^X| + |\text{Relations}^Y|} \quad (32)$$

where  $\text{Relations}^{X/Y}$  is the set of direct links in network X/Y and  $\text{Correlations}$  is their intersection.

## 4.3 Experimental Results

We compare the performance of IONE and its variations with the baselines. In our experiment, we set 90% of the anchor users as the training set and the rest as the test set.

3. The data and code are available in <https://github.com/ColaLL/AcrossNetworkEmbeddingDiversity>

### 4.3.1 Performance of IONES without soft constraints

We first compare the performance of IONES with only hard constraints considered with the baselines.

The experimental results based on the TF dataset are presented in Fig. 4. IONE outperforms most of the baselines significantly given different @N settings as well as different training-to-test ratios, and has similar performance as BootEA. And IONE-Con outperforms all the methods. In particular, MAH and MAG performs better than CRW, showing that the *directed* random walk approach is not as accurate as the manifold alignment approach. Note that, different from our proposed method, the methods based on random walks (e.g., CLF and CRW) are “asymmetric” in which taking different network as the source network for a particular network pair will lead to different prediction results. For such methods, the alignment has to be computed in both directions, and thus the computation complexity doubles. Besides, both MAG and MAH fail to differentiate the follower-ship and followee-ship when constructing the incidence matrices of the hypergraph. PALE\_DeepWalk which incorporates network embedding and random walk approach outperforms PALE\_LINE as it considers the indirected context during the walk process. Without the attribute feature, ULINK does not perform well when only the structural feature is available. We found that even INE and ONE can give better results. For the DBLP dataset, similar performance comparison results are obtained (Fig.5). In addition, we have also compared the performance of IONE and IONE-Con with that of FRUI. We use P@1 only as this is the metric being used in the paper proposing FRUI. For the TF dataset, FRUI can achieve P@1 = 0.1329 while IONE and IONE-Con can achieve P@1 = 0.2057 and P@1 = 0.3481 respectively. For the DBLP dataset, FRUI can achieve P@1 = 0.2867 while IONE and IONE-Con can go up to P@1 = 0.2096 and P@1 = 0.3088.

For different training-to-test ratios, as observed from Fig.4(b) and Fig.5(b), IONE outperforms most of the baseline methods and IONE-Con gives the best performance. Even for ratio settings as low as 10%, 20%, the performance enhancement is still significant. We observe that when the training ratio is very low, the PALE\_DeepWalk performs better than other baselines. One possible reason is that when the number of training anchors is small, indirect contexts which is used in PALE\_DeepWalk are indeed helpful for learning distinguished embeddings.

Among the subspace learning based methods, we also

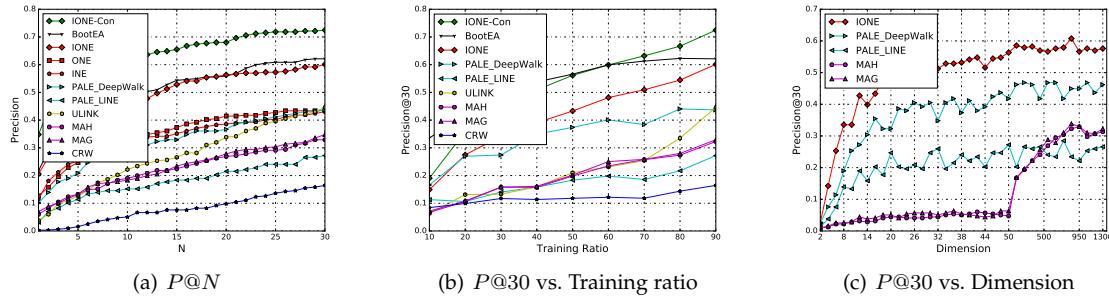


Fig. 4. Detailed Performance Comparison on Twitter-Foursquare Dataset

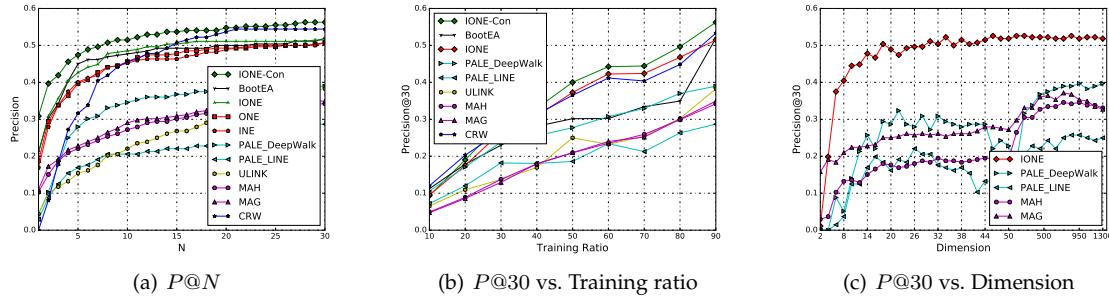


Fig. 5. Detailed Performance Comparison on DBLP Dataset

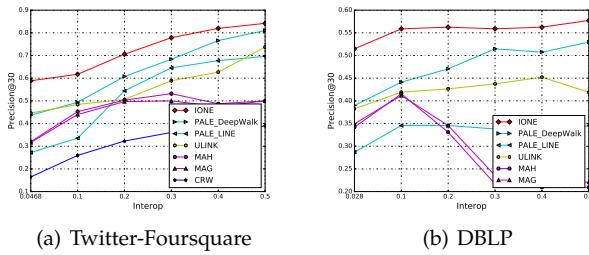


Fig. 6.  $P@30$  vs. Interop

compare their performance under the settings using representations of different dimensions<sup>4</sup> for details). According to Figs. 4(c) and 5(c), both *MAG* and *MAH* achieve their best performance when the dimensionality setting is around 950. And PALE models show a not-stable tendency in this dimension analysis where they reach an accepted performance when the dimension is low on the TF dataset, but need a high dimension (larger than 500) on the DBLP dataset. *IONE* reaches its best performance when the dimensionality is under 50. It is well known that the complexity of the learning algorithm is highly depending on the dimensionality of the subspace. Besides, low-dimensional representation also leads to an efficient relevance computation. Our proposed network embedding approach is also significantly more efficient than the matrix factorization-based approach.

Figs. 6(a) and 6(b) show how different methods perform

4. We exclude ULINK as the memory of this model used exceeds our platform when the dimension is higher than 40, where more than 36G memory costed when dimension is 30. Note that the precisions of *IONE* and its variants are still higher than ULINK even the dimension of these model is 30 (refer to Fig.4(c),5(c),7(c) and 8(c))

at different values of *interop*<sup>5</sup>. Intuitively, it will be easier to align the network when the networks share more common links. We observe that most of methods achieve better result as the *interop* value increases, except for *MAH* and *MAG*. Their precision values increase first and then decrease as the *interop* value increases.

#### 4.3.2 Effect of Incorporating Structural Diversity

In this section, we investigate the effectiveness of incorporating the proposed diversity measure into the *IONE* framework. We compare the performance results obtained by the proposed models, *IONE-Ex*, *IONE-D-Ex* and *IONE-Con-Ex*. Figs. 7 and 8 show the experimental results, from which we obtain the following findings:

1) In general, *IONE-D-Ex* outperforms *IONE-Ex* for  $P@N$  at different  $N$  settings as well as different training-to-test ratios. The observation is consistent with our previous conjecture that the structural diversity will boost the performance of the alignment. But we argue that *IONE-D* considers the two-step path  $v_i^X \rightleftharpoons v_a \rightleftharpoons v_i^X$ , and only the nodes connecting to anchors take part in the learning process as positive samples, which may lead to user pairs who share more diverse first-order neighbor anchors to be selected as aligned pairs preferentially. While the *IONE* model considers all the nodes across networks in the learning process. Even for two nodes without any direct connection to any anchors, their embeddings can also be “similar” to some degree when their neighbors have “similar” embeddings, as their neighbors may share enough anchors, so on so forth. *IONE-Con* concatenates the representations obtained from *IONE* and *IONE-D* and inherits their intrinsic advantages.

5. In our experiments, we varies the interop value by removing non-anchor relations in the networks.

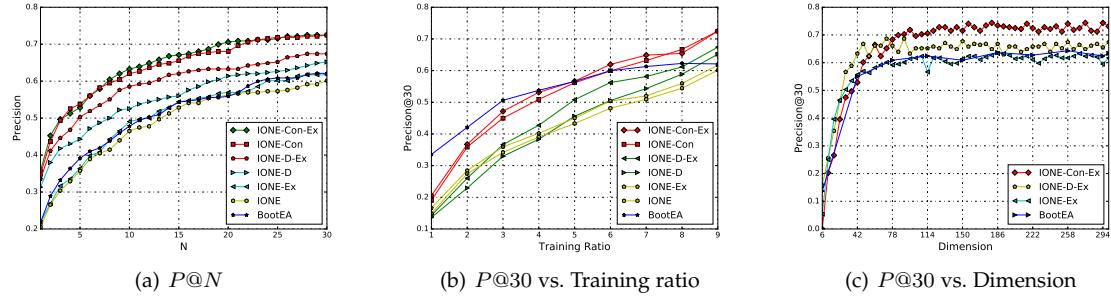


Fig. 7. Community Diversity Embedding Performance Comparison on Twitter-Foursquare Dataset

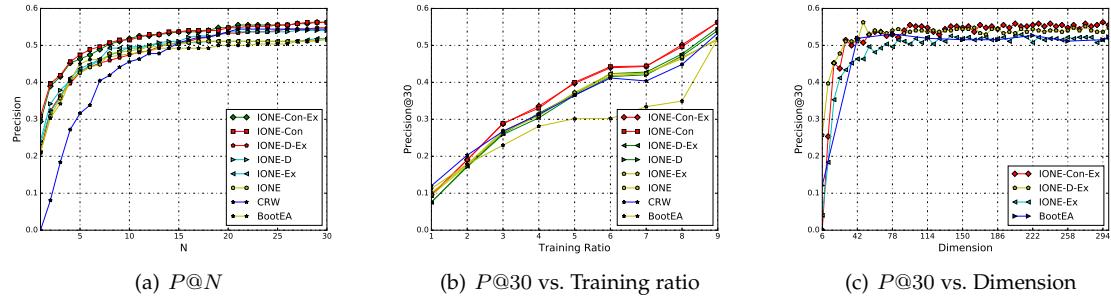


Fig. 8. Community Diversity Embedding Performance Comparison on DBLP Dataset

Thus, *IONE-Con* so far achieve the best performance and outperforms all the baselines consistently and significantly given different @ $N$  settings.

2) The use of the extended representations, that is including also the input and output context vectors, is useful to achieve better performance (e.g., comparing *IONE-Ex* vs. *IONE* and *IONE-D-Ex* vs. *IONE-D* in Fig.7(a)). To explain this result, we refer to the definition of input and output context vectors in Eq.(23).  $v_i^X$ 's output context vector  $\vec{u}_i^{X'X}$  is updated by its follower's node vector  $\vec{u}_j^X$  (and  $v_k^Y$ 's follower  $v_j^X$ 's node vector  $\vec{u}_j^{Y'Y}$  when  $p_a(v_i^X|v_k^Y)$  acts as "bridge" between  $v_k^Y$  and  $v_i^X$ . In this case,  $\vec{u}_j^{Y'Y}$  also can be seen as  $v_i^X$ 's follower's node vector with probability  $p_a(v_i^X|v_k^Y)$ ). Similar with Eq. (23), the updating of  $v_j^X$ 's input context vector  $\vec{u}_j^{X'X}$  is based on its follower's node vector (which is illustrated in Eq.(24)). Following these updating rules, ones' output (input) context vectors will be close in the embedding space when they share more followers (followers) as the output (input) context vectors are updated by its' followers' (followers') node vectors. Based on this, we can conclude that output (input) context vectors can be used as additional information to enhance the performance of alignment as they preserve a follower (follower) structural proximity across networks. We believe that this is the reason why *IONE-(D)-Ex* model achieves a better performance. However, for the DBLP dataset, similar enhancement is not observed (see Fig.8(a)), which hints that the input and output context vectors do not contain much additional information. This is probably due to the way the DBLP networks are constructed.

3) We also investigate the effect of dimensionality to the *IONE* framework. Figs. 7(c) and 8(c) show the performances obtained by *IONE*'s variants under the same settings of the

dimensionality. And we observe that all the models tend to be relatively stable when the dimensionality is larger than 150, and that *IONE-Con-Ex* performs the best among all the models. When the dimensionality is smaller than 150, *IONE-Con-Ex* is slightly worse than *IONE-D-Ex* with the same dimensionality setting on the DBLP dataset. That is because *IONE-Con-Ex* concatenates the representations from *IONE-Ex* and *IONE-D-Ex* under the settings of half size of the compared dimensionality. For example, *IONE-Con-Ex* (Dim=100) concatenates the representations obtained from *IONE-Ex* (Dim=50) and *IONE-D-Ex* (Dim=50), and it is obvious to see that neither of *IONE-Ex* and *IONE-D-Ex* are well trained when the dimensionality is set as 50. However the concatenation empowers *IONE-Con-Ex* to achieve a performance slightly lower than *IONE-D-Ex* (Dim=100) but far better than *IONE-Ex* (Dim=50) and *IONE-D-Ex* (Dim=50). Thus, when *IONE-Ex* and *IONE-D-Ex* are better trained by increasing the dimensionality (Dim=75), *IONE-Con-Ex* outperforms all the baselines (after the point of Dim=150).

#### 4.3.3 Effect of Adding Soft Constraints

To evaluate the effect of incorporating soft constraints into the *IONE* framework, we compare the performance of *IONE-S*, *IONE* and other baselines. In *IONE-S*, the soft constraint  $p_a(v_i^X|v_k^Y)$  was derived similar to what being proposed in *CLF* [11]. By considering the labeled anchor pairs as positive data and those coming from randomly sampled unlabeled user pairs across networks as negative data, we compute the structural features including common neighbors, extended Jaccard's coefficient and extended Adamic/Adar measure for a Logistic Regression classification model (referred as LR-WithoutDiversity) to estimate  $p_a(v_i^X|v_k^Y)$  for *IONE-S*. For *IONE-D-S*, we add community diversity related features (community diversity values of neighbors,

Adamic/Adar measure weighted by diversity values) to the classifier (referred as LR-WithDiversity). With the estimated  $p_a(v_i^X | v_k^Y)$  incorporated as the soft constraints, the network embeddings are then obtained via IONE-S and IONE-D-S respectively. In general, the classification model achieves better performance when the training sets are more balanced. Here we use imbalance ratio  $\frac{|\text{-ve anchor links}|}{|\text{+ve anchor links}|}$  as a proxy to reflect the performance of the empirical classification model.

Fig.9(a) and 9(b) show that IONE-S outperforms IONE, and IONE-D-S outperforms IONE-D over all the imbalance ratio settings, and especially when the imbalance ratio is high. This shows the effectiveness of adding the soft constraints. As anticipated, IONE-D-S outperforms IONE-D-S due to the consideration of the diversity measure as previously discussed. Also, both IONE-S and IONE-D-S outperform all the other baselines.

Regarding the convergence of the model inference, Fig. 9(c) shows that IONE-S and IONE both converge in a stable manner, but IONE-S and IONE-D-S achieve its convergence much earlier than IONE. We believe that it is because there is an edge sampling involved in the learning process (See in Algorithm 1). For learning IONE-D, an edge re-forming process is involved (see Algorithm 2). The empirical soft constraints will lead to the number of re-formed edges to be much bigger than the number of the original network.

#### 4.4 Case Study

Fig.10 illustrates subgraphs of two social networks utilized in our experiments with the ground truth of the alignment, and their embeddings in the inferred common subspace. We adopt t-SNE [41] for visualizing the embeddings. The red, green and blue nodes in the two subgraphs denote the anchor users in the training set, the anchor users in the test set, and the users only belonging to one network respectively. Generally speaking, topologically similar nodes with the help of the clue provided by the anchor links are projected to locations close to each other in the embedded space. In particular, we observe that:

- **Two nodes sharing more common edges in one network appear closer in the learned low dimensional space.** In the Twitter network, *Bar\_tw* shares with *jac\_tw* two input edges from *kyl\_an* and *hue\_an*, but none with *JES\_tw* which in turn has an input edge to *kyl\_an*. Thus *Bar\_tw* appears closer to *jac\_tw* than *JES\_tw* in the embedded space (near the left side).
- **The anchor links do help the network alignment.** We observe that *jam\_fs* has 3 input edges and 3 output edges from 3 anchor users (*kyl\_an*, *hue\_an*, *mil\_an*) in the Foursquare network, while *jam\_tw* also has 3 input and 2 output edges from the same set of anchor users in the Twitter network. In the plot of the embedded space (near the lower side), *jam\_tw* (green octagon) and *jam\_fs* (green cross) are located very close to each other.
- **User proximity is preserved in the embedded space based on both follower-ship and followee-ship collaboratively.** Consider *jes\_fs*, *rad\_fs* and *tim\_fs* in the Foursquare network. All three have edges connecting to *kyl\_an*. Node *jes\_fs* has a bi-directed edge with *kyl\_an*. Nodes *rad\_fs* and *tim\_fs* have only input edges

from *kyl\_an*. Therefore, *rad\_fs* and *tim\_fs* are closer to each other in the inferred space (near upper right corner) and farther apart from *jes\_fs*. For another example, *JES\_tw* is the only node in the Twitter network pointing to *kyl\_an*, and *jes\_fs* is one of two nodes in the Foursquare network pointing to *kyl\_an*. Thus, *kyl\_an* has significant contribution to both *jes\_fs* and *JES\_tw*, drawing them close in the inferred space (near lower left corner).

Moreover, we have found some instances, such as *cmuiz\_twitter* and *cmuiz\_foursquare* which do not share any anchors in the dataset, still are predicted correctly by IONE at  $P@1$ . Recall Eq.(21) and (22), we can see that a target node vectors ( $\vec{u}_i^X$  and  $\vec{u}_j^X$ ) are mainly updated by its' neighbors' input and output context vectors (negative samples can be ignored in this deduction as they are mainly used for pulling away the distance between temp and target node). Then we can conclude that two users should be close in the embedded space when they share enough "similar" neighbors as input and output context vectors. Furthermore, we can obtain the conclusion that the more coinciding anchor (which is optimized in objective 2 of IONE) context vectors two users share, more "similar" embedding these two users have. Based on this conclusion, we define a metric *anchor shared ratios* between two users as the number of shared anchors divided by the number of total neighbors of these two users for this case analysis. And we've found that the *anchors shared ratios* of *cmuiz\_twitter* and *cmuiz\_foursquare*'s neighbors are 0.064 and 0.112 respectively, which is significantly higher than the average value (0.004 and 0.001) of all user pairs across networks. It is consistent with our previous guess that testing instances sharing no anchors can still be aligned by IONE when they have "similar" neighbors who share enough anchors.

Besides, we've also conducted case studies to test the efficacy of IONE-D and verified the importance of the diversity indicator to the improvement over IONE. For example, in twitter-foursquare dataset, *datachick\_foursquare* shares 13 anchor friends with other three users (*AstroN8\_twitter*, *yan-cysco\_twitter* and *datachick\_twitter*). In the meanwhile, *datachick\_foursquare* share 14 anchor friends with another two users (*shimpster\_twitter* and *Catahouligan\_twitter*). And the ground truth is that *datachick\_foursquare* and *datachick\_twitter* form the aligned pair. However, IONE failed to find such pair, as IONE tends to rank high those nodes with higher ratio of common friends. And in this case, only *shimpster\_twitter* and *Catahouligan\_twitter* squeeze in the obtained top 5 candidate list by IONE. However, with the help of the diversity indicator as illustrated in Sec.3.3, and in this case, the value of the diversity measurement between *datachick\_foursquare* and *datachick\_twitter* is 3, whereas the diversity between *datachick\_foursquare* and other users are all 1, thus *datachick\_twitter* rank highest in the candidate list obtained by IONE-D which contributes to the  $P@1$  accuracy.

## 5 CONCLUSION

In this paper, we studied the problem of mapping users across networks. A representation learning model with the

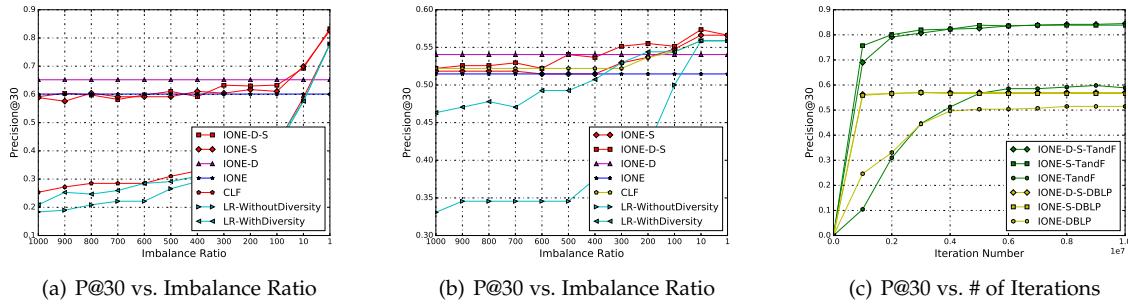


Fig. 9. Performance Comparison for IONE and IONE-D with Soft Constraints

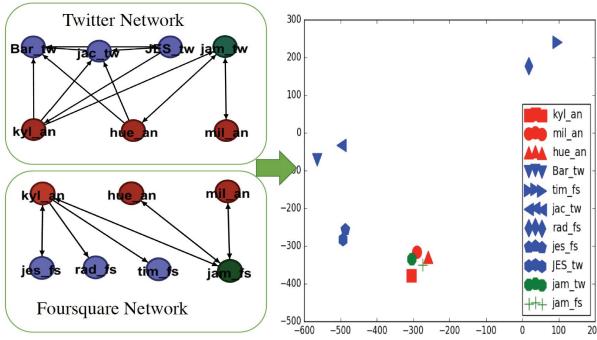


Fig. 10. Visualizing a Subgraph of Tw-Fq. in the Projected Space.

objective to learn an aligned network embedding for multiple networks was proposed. The proposed approach explicitly models the follower-ship and followee-ship of each user as the input and output context. Both given and potential anchor links can be used in this model as hard and soft constraints in a unified framework for learning. Stochastic gradient descent and negative sampling are used for the efficient learning of the model. We further incorporate the structural diversity based embedding into the user embeddings for a precisely user representation. The empirical results obtained based on extensive experiments conducted on two real-world datasets demonstrate that our proposed model outperforms several state-of-the-art methods.

Even though the proposed models have a better performance, there still exist some limitations requiring future research efforts. Firstly, for *IONE-S*, soft constraints are obtained by a pre-trained classifier. There will be a high imbalance ratio between the positive and negative instances when the anchor set is small. A more efficient and effective way for obtaining the soft constraints can be needed. Secondly, *IONE-D* cannot be easily combined into *IONE-S*.

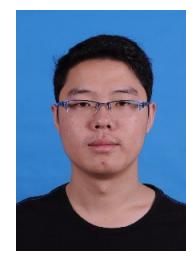
## ACKNOWLEDGMENT

The work is partially supported by National Key R&D Program of China under Grant No.2017YFB0802305 and NSFC Grant 61772074 and 61806031, and in part by Chongqing Research Program of Basic Research and Frontier Technology under Grants cstc2017jcyjAX0325, cstc2017jcyjAX0406, and in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grants KJQN201800638.

## REFERENCES

- [1] Y. Dong, J. Tang, S. Wu, J. Tian, N. V. Chawla, J. Rao, and H. Cao, "Link prediction and recommendation across heterogeneous social networks," in *Proceedings of the IEEE 12th International Conference on Data Mining*. Brussels, Belgium: IEEE, December 2012, pp. 181–190.
- [2] J. Zhang, P. S. Yu, and Z.-H. Zhou, "Meta-path based multi-network collective link prediction," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, USA: ACM, August 2014, pp. 1286–1295.
- [3] J. Zhang, X. Kong, and P. S. Yu, "Transferring heterogeneous links across location-based social networks," in *Proceedings of the 7th ACM international conference on Web search and data mining*. New York City, USA: ACM, February 2014, pp. 303–312.
- [4] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu, "Personalized recommendation via cross-domain triadic factorization," in *Proceedings of the 22nd international conference on World Wide Web*, Seoul, Korea, April 2013, pp. 595–606.
- [5] T. Man, H. Shen, J. Huang, and X. Cheng, "Context-adaptive matrix factorization for multi-context recommendation," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. Melbourne, Australia: ACM, October 2015, pp. 901–910.
- [6] R. Zafarani and H. Liu, "Connecting corresponding identities across communities," in *Proceedings of the Third International Conference on Weblogs and Social Media*, San Jose, California, USA, May 2009, pp. 354–357.
- [7] J. Liu, F. Zhang, X. Song, Y. Song, C. Lin, and H. Hon, "What's in a name?: an unsupervised approach to link users across communities," in *Proceedings of the Sixth International Conference on Web Search and Data Mining*. Rome, Italy: ACM, February 2013, pp. 495–504.
- [8] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "COSNET: connecting heterogeneous social networks with local and global consistency," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Sydney, NSW, Australia: ACM, August 2015, pp. 1485–1494.
- [9] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "HYDRA: large-scale social identity linkage via heterogeneous behavior modeling," in *Proceedings of International Conference on Management of Data*. Snowbird, UT, USA: ACM, June 2014, pp. 51–62.
- [10] R. Zafarani and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Minin*. Chicago, IL, USA: ACM, August 2013, pp. 41–49.
- [11] J. Zhang and P. S. Yu, "Integrated anchor and social link predictions across social networks," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, July 2015, pp. 2125–2132.
- [12] S. Wu, H. Chien, K. Lin, and P. S. Yu, "Learning the consistent behavior of common users for target node prediction across social networks," in *Proceedings of the 31th International Conference on Machine Learning*, Beijing, China, June 2014, pp. 298–306.
- [13] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen, "Mapping users across networks by manifold alignment on hypergraph," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Québec City, Québec, Canada, July 2014.
- [14] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical*

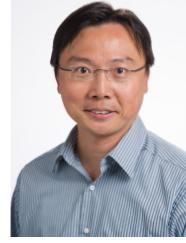
- Mechanics Theory and Experiment*, vol. 2008, no. 10, pp. 155–168, 2008.
- [15] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York City, USA: ACM, August 2014, pp. 701–710.
- [16] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: large-scale information network embedding,” in *Proceedings of the 24th International Conference on World Wide Web*, Florence, Italy, May 2015, pp. 1067–1077.
- [17] G. W. Klau, “A new graph-based method for pairwise global network alignment,” *BMC bioinformatics*, vol. 10, no. Suppl 1, p. S59, 2009.
- [18] D. Koutra, H. Tong, and D. Lubensky, “Big-align: Fast bipartite graph alignment,” in *Proceedings of IEEE 13th International Conference on Data Mining*. Dallas, Texas, USA: IEEE, December 2013, pp. 389–398.
- [19] J. Zhang and P. Yu, “Multiple anonymized social networks alignment,” in *Proceedings of IEEE 15th International Conference on Data Mining*, Atlantic City, NJ, USA, November 2015, pp. 389–398.
- [20] J. Zhang and P. S. Yu, “Pct: Partial co-alignment of social networks,” in *Proceedings of the 25th World Wide Web Conference*, ser. WWW ’16, 2016.
- [21] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, “Linking users across domains with location data: Theory and validation,” in *International Conference on World Wide Web*, 2016, pp. 707–719.
- [22] X. Zhou, X. Liang, X. Du, and J. Zhao, “Structure based user identification across social networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1178–1191, 2018.
- [23] M. Heimann, H. Shen, T. Safavi, and D. Koutra, “Regal: Representation learning-based graph alignment,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 117–126.
- [24] Z. Zhong, Y. Cao, M. Guo, and Z. Nie, “Colink: An unsupervised framework for user identity linkage,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [25] Y. Nie, Y. Jia, S. Li, X. Zhu, A. Li, and B. Zhou, “Identifying users across social networks based on dynamic core interests,” *Neurocomputing*, vol. 210, pp. 107–115, 2016.
- [26] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, “Identifying users across social tagging systems,” in *International Conference on Weblogs and Social Media*, Barcelona, Catalonia, Spain, July, 2011.
- [27] J. Vosecky, D. Hong, and V. Y. Shen, “User identification across multiple social networks,” in *International Conference on Networked Digital Technologies*, 2009, pp. 360–365.
- [28] S. H. K. Parthasarathi, S. H. K. Parthasarathi, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira, “Exploiting innocuous activity for correlating users across sites,” in *International Conference on World Wide Web*, 2013, pp. 447–458.
- [29] O. Peled, M. Fire, L. Rokach, and Y. Elovici, “Entity matching in online social networks,” in *International Conference on Social Computing*, 2014, pp. 339–344.
- [30] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manils, “How unique and traceable are usernames?” vol. 6794, pp. 1–17, 2011.
- [31] F. Buccafurri, G. Lax, A. Nocera, and D. Ursino, “Discovering links among social networks,” in *Ecmi/pkdd*, 2012, pp. 467–482.
- [32] X. Kong, J. Zhang, and P. S. Yu, “Inferring anchor links across multiple heterogeneous social networks,” in *Proceedings of the 22nd ACM international conference on information and knowledge management*. San Francisco, CA, USA: ACM, October 2013, pp. 179–188.
- [33] X. Mu, F. Zhu, J. Wang, J. Wang, J. Wang, and Z. H. Zhou, “User identity linkage by latent user space modelling,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1775–1784.
- [34] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, “Bootstrapping entity alignment with knowledge graph embedding,” in *International Joint Conference on Artificial Intelligence*, 2018, pp. 4396–4402.
- [35] W. Zhao, S. Tan, Z. Guan, B. Zhang, M. Gong, Z. Cao, and Q. Wang, “Learning to map social network users by unified manifold alignment on hypergraph,” *IEEE Transactions on Neural Networks and Learning Systems*, no. 99, pp. 1–13, 2018.
- [36] T. Man, H. Shen, S. Liu, X. Jin, and X. Cheng, “Predict anchor links across social networks via an embedding approach,” in *International Joint Conference on Artificial Intelligence*, 2016, pp. 1823–1829.
- [37] J. Zhang, B. Chen, X. Wang, H. Chen, C. Li, F. Jin, G. Song, and Y. Zhang, “Mego2vec: Embedding matched ego networks for user alignment across social networks,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 327–336.
- [38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems 2013*, Lake Tahoe, Nevada, USA, December 2013, pp. 3111–3119.
- [39] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, “Reducing the sampling complexity of topic models,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 891–900.
- [40] X. Zhou, X. Liang, H. Zhang, and Y. Ma, “Cross-platform identification of anonymous identical users in multiple social media networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 411–424, 2016.
- [41] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 2579–2605, p. 85, 2008.



**Li Liu** received the Ph.D. in computer science from Beijing Institute of Technology in 2016, the M.E. in computer science from Kunming University of Science and Technology in 2012 and the B.B.A. in Information management and information system from Chongqing University of Posts and Telecommunications in 2009. He was a visiting student at Hong Kong Baptist University during 2016. He is currently an Assistant Professor in Chongqing University of Posts and Telecommunications. His research interests include web mining and social computing.



**Xin Li** is currently an Associate Professor in the School of Computer Science at Beijing Institute of Technology, China. She received the B.Sc. and M.Sc degrees in Computer Science from Jilin University China, and the Ph.D. degree in Computer Science at Hong Kong Baptist University. Her research focuses on the development of algorithms for representation learning, reasoning under uncertainty and machine learning with application to Natural Language Processing, Recommender Systems, and Robotics.



**William K. Cheung** received the B.Sc. and M.Phil. degrees in electronic engineering from the Chinese University of Hong Kong, Hong Kong, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 1999. He is currently the Head and Associate Professor of the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include artificial intelligence, data mining, collaborative information filtering, social network analysis, and healthcare informatics. Dr. Cheung has served as the Co-Chair and a Program Committee Member for a number of international conferences and workshops, as well as a Guest Editor of journals on areas, including artificial intelligence, Web intelligence, data mining, Web services, e-commerce technologies, and health informatics. Since 2002, he has been on the Editorial Board of the IEEE Intelligent Informatics Bulletin.



**Lejian Liao** received a Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences. He is currently a professor in the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. He also served as Vice Dean of the school. With main research interest in machine learning, natural language processing and intelligent network, Professor Lejian Liao has published numerous papers in several areas of computer science.