Aligning Users Across Social Networks Using Network Embedding

Li Liu,[1] William K. Cheung,[2] Xin Li,[1*] and Lejian Liao[1]
[1] BJ ER Center of HVLIP&CC, School of Comp. Sci., Beijing Institute of Technology, Beijing, China
[2] Department of Computer Science, Hong Kong Baptist University, Hong Kong, China

Abstract

In this paper, we adopt the representation learning approach to align users across multiple social networks where the social structures of the users are exploited. In particular, we propose to learn a network embedding with the followership/followee-ship of each user explicitly modeled as input/output context vector representations so as to preserve the proximity of users with "similar" followers/followees in the embedded space. For the alignment, we add both known and potential anchor users across the networks to facilitate the transfer of context information across networks. We solve both the network embedding problem and the user alignment problem simultaneously under a unified optimization framework. The stochastic gradient descent and negative sampling algorithms are used to address scalability issues. Extensive experiments on real social network datasets demonstrate the effectiveness and efficiency of the proposed approach compared with several state-of-the-art methods.

1 Introduction

Mapping users across online social networks has recently been attracting attention in both academia and industry. The established user correspondence can benefit applications like social link prediction [Dong *et al.*, 2012; Zhang *et al.*, 2014] and cross-domain recommendation [Hu *et al.*, 2013]. In general, carefully aligning heterogeneous social networks can alleviate the sparsity issue and transfer useful information for social network analysis.

One intuitive way to establish the user correspondence is to make use of user demographic attributes like username, gender, and etc. Among them, username is most commonly used with the argument that users tend to use their favorite usernames or related variants for multiple networks. The effectiveness of this simple approach has been demonstrated under some given experiment settings [Liu *et al.*, 2013; Zhang *et al.*, 2015]. Yet, there exist users who deliberately use different usernames. In addition, 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 [Liu *et al.*, 2014; Zafarani and Liu, 2013].

Alternatively, the structural information of the social networks can be used directly for user alignment. Intra-links, inter-links and common users across the networks (also called anchor users) can be exploited to derive a probabilistic graph classifier [Zhang and Yu, 2015a; Wu *et al.*, 2014] or to render a common subspace of multiple networks for relevance computation [Tan *et al.*, 2014]. Most of these related work considers the links to be undirected. However, follower-followee relations are often maintained in a number of social media like Twitter. 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 social interest. It is intuitive that the follower-ship and followee-ship collaboratively define one's unique social figure in virtual networks. Also, most of the existing work computes the structural alignment of networks using matrix factorization where matrix inverse is typically involved, making them hard to scale up for large-scale problems.

In this paper, we propose to align social networks using the network representation learning (NRL) approach [Perozzi *et al.*, 2014; Tang *et al.*, 2015b; 2015a]. We extend it to social network alignment using a unified optimization framework where the embeddings of multiple networks are learned simultaneously subject to hard and soft constraints on common users of the networks. To contrast with the existing NRL methods, the proposed network embedding model explicitly represents the follower-ship and followee-ship of each user as the input and output context vectors. We name it as the *Input-Output Network Embedding* (IONE). The IONE can preserve the proximity of users with "similar" sets of followers and followees in the embedded space. Also, both known and potential anchor users across the networks can be introduced in a unified manner to play the roles as hard and soft constraints for facilitate the transfer of the contextual information. All the considering factors are formulated into a single objective function so that minimizing it can allow network embedding learning and user alignment to be achieved at the same time. Also, we adopt the negative sampling which can substantially

*Corresponding Author: Xin Li (xinli@bit.edu.cn).

reduce the computation cost and make the IONE scale well to networks of large size. We conduct detailed empirical evaluation using real datasets and show that the IONE outperforms other competitive approaches.

2 Related Work

Structural alignment methods proposed for social network alignment can roughly be categorized into *supervised* and *unsupervised*. The unsupervised methods assume absence of anchor links, making the problem essentially a generic graph alignment problem [Koutra *et al.*, 2013; Zhang and Yu, 2015b; 2016]. The supervised methods predict anchor links using the classification approach based on features like username, profile information, and social connectivity measures [Malhotra *et al.*, 2012; Kong *et al.*, 2013]. Recently, fusing anchor link prediction and social link prediction was also found effective [Zhang and Yu, 2015a].

Learning to align the network manifolds is another promising direction. In [Tan *et al.*, 2014], hyper-edges among the users are defined and the manifolds of two social networks are projected onto a common embedded space so that the nodes in each hyper-edge will be close in the projected space. Eigenvalue decomposition on a matrix defined based on the hyperedges is required, which however is inevitably time-consuming. Our work is along the direction of projecting multiple social networks onto a common embedded space. To contrast, the proposed IONE allows more social structural properties to be exploited for the network representation learning, with the ultimate goal to result in a more accurate social network alignment. The IONE does not require explicit definition of relations as hyperedges but can achieve a similar goal implicitly via the interactions of the input and output context representations of the nodes in the network.

3 Model Framework

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

3.1 Input-Output Network Embedding

We propose a novel network embedding for representing social networks. Similar to most of the existing representation learning methods, we represent each node $v_i \in V$ as a d-dimensional vector in an embedded space via a projection function $f : V \to \Re^d$. By leveraging the follower-followee relations of the users, we allow the context of each user in a social network to be characterized by its own 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 $\vec{u_i} \in \Re^d$, an input context vector $\vec{u_i}' \in \Re^d$, and an output context vector $\vec{u_i}'' \in \Re^d$. As illustrated in Fig.1, v_i is the parent node of v_j and thus $\vec{u_i}$ contributes to $\vec{u_j}'$ which represents the input context 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).

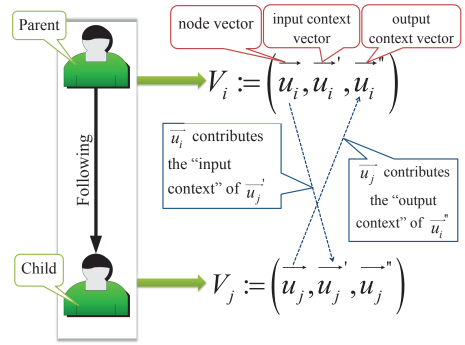


Figure 1: An Example of User Vector Representations

In order to learn the network embedding from a social network G, for each edge $(v_i, v_j) \in E$, we define the probability that v_i contributes specifically to (or "generates" as termed in [Tang *et al.*, 2015b]) v_j as its *input context* when compared with how v_i contributes to other nodes, given as

$$p_1(v_j|v_i) = \frac{\exp(\vec{u_j}'^T \cdot \vec{u_i})}{\Sigma_{k=1}^{|V|} \exp(\vec{u_k}'^T \cdot \vec{u_i})} \quad (1)$$

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 other nodes, given as

$$p_2(v_i|v_j) = \frac{\exp(\vec{u_i}''^T \cdot \vec{u_j})}{\Sigma_{k=1}^{|V|} \exp(\vec{u_k}''^T \cdot \vec{u_j})} \quad (2)$$

We further define the empirical counterparts of $p_1(v_j|v_i)$ and $p_2(v_i|v_j)$ as $\widehat{p}_1(i,j) = w_{ij}/d_i^{out}$ and $\widehat{p}_2(i,j) = w_{ij}/d_j^{in}$ 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 $\widehat{p_1}$ and $\widehat{p_2}$, the corresponding IONE model can be inferred (See Sec.3.2).

3.2 Aligning Network Embeddings of Multiple Networks

Given two social networks X and Y, we propose to compute their IONE models and at the same time align them. To do that, 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 in the sequel. 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 coincide in the embedded space and those who are close in the embedded space can be considered as good candidates for user alignment.

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

$$\begin{aligned} O_1 &= -\sum_{k\in\{X,Y\}}\sum_{v_j\in V^k}\lambda_i^{out}KL(\widehat{p}_1^k(i,j)||p_1(v_j^k|v_i^k)) \\ &\quad -\sum_{k\in\{X,Y\}}\sum_{v_i\in V^k}\lambda_j^{in}KL(\widehat{p}_2^k(i,j)||p_2(v_i^k|v_j^k)). \quad (3)\end{aligned}$$

By setting λ_i^{out} as the output-degree d_i^{out} of v_i and λ_j^{in} as the input-degree d_j^{in} of v_j, the objective function can be rewritten as:

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

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 the alignment accuracy for nodes other than the anchor ones, we train a classifier for anchor link prediction and use the anchor link prediction results as "soft" constraints. 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 I). 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_k^Y in network Y can contribute as input context to v_i^X in network X with the probability $p_a(v_i^X|v_k^Y)$. Then, v_k^Y in network Y can contribute to other nodes 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 the empirical probabilities $\hat{p}_1(v_j^{X/Y}|v_k^{Y/X}) = \sum_{v_i\in X/Y} p_a(v_i^{X/Y}|v_k^{Y/X}) * w_{ij}/d_i^{out}$, $\hat{p}_2(v_i^{X/Y}|v_k^{Y/X}) = \sum_{v_j\in X/Y} p_a(v_j^{X/Y}|v_k^{Y/X}) * w_{ij}/d_j^{in}$. We can again minimize the KL-divergence of p and the empirical distribution $\widehat{p}$, and the corresponding objective function becomes:

$$\begin{aligned} O_2 &= -\sum_{v_k\in Y}\sum_{(v_i,v_j)\in E^X} w_{ij}^X p_a(v_i^X|v_k^Y)\log p_1(v_j^X|v_k^Y) \\ &\quad -\sum_{v_k\in Y}\sum_{(v_i,v_j)\in E^X} w_{ij}^X p_a(v_j^X|v_k^Y)\log p_2(v_i^X|v_k^Y) \\ &\quad -\sum_{v_k\in X}\sum_{(v_i,v_j)\in E^Y} w_{ij}^Y p_a(v_i^Y|v_k^X)\log p_1(v_j^Y|v_k^X) \\ &\quad -\sum_{v_k\in X}\sum_{(v_i,v_j)\in E^Y} w_{ij}^Y p_a(v_j^Y|v_k^X)\log p_2(v_i^Y|v_k^X) \quad (5)\end{aligned}$$

Thus, the multiple networks can be aligned in the embedded space by minimizing the combined objective function $O = O_1 + O_2$ over the parameters $\{(\overrightarrow{u_x}^X, \overrightarrow{u_x}'^X, \overrightarrow{u_x}''^X)\}$ and $\{(\overrightarrow{u_y}^Y, \overrightarrow{u_y}'^Y, \overrightarrow{u_y}''^Y)\}$ 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 embeded space, and O_2 allows the contexts of the anchor nodes to be propapated across the networks which in turn can enhance the individual learning of the two embeddings.

3.3 Model Inference

We use the stochastic gradient descent to learn the vector representations of the two networks. To update the node vector of v_i in network X, i.e., $\overrightarrow{u_i}^X$, the gradient is computed as:

$$\begin{aligned} \frac{\partial O}{\partial \overrightarrow{u_i}^X} &= \frac{\partial O_1}{\partial \overrightarrow{u_i}^X} + \frac{\partial O_2}{\partial \overrightarrow{u_i}^X} = w_{ij}^X * \frac{\partial \log p_1(v_j^X|v_i^X)}{\partial \overrightarrow{u_i}^X} + \\ &\quad \sum_{v_k\in V^X} w_{ij}^Y * p_a(v_i^Y|v_k^X)\frac{\partial \log p_1(v_j^Y|v_k^X)}{\partial \overrightarrow{u_i}^X} \quad (6)\end{aligned}$$

Calculating the conditional probability p_1 in Eq.(6) requires the summation over the entire set of nodes. To reduce the computational complexity, we adopt the negative sampling approach [Mikolov *et al.*, 2013]. In short, it 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, given as:

$$\begin{aligned} \log p_1(v_j^X|v_i^X) &\propto \log\sigma(\overrightarrow{u_j}'^{X^T}\cdot\overrightarrow{u_i}^X) \quad (7) \\ &\quad +\Sigma_{m=1}^K E_{v_n\sim p_n(v)}\log\sigma(-\overrightarrow{u_n}'^{X^T}\cdot\overrightarrow{u_i}^X)\end{aligned}$$

$$\begin{aligned} \log p_1(v_j^Y|v_k^X) &\propto \log\sigma(\overrightarrow{u_j}'^{Y^T}\cdot\overrightarrow{u_k}^X) \quad (8) \\ &\quad +\Sigma_{m=1}^K E_{v_n\sim p_n(v)}\log\sigma(-\overrightarrow{u_n}'^{Y^T}\cdot\overrightarrow{u_k}^X)\end{aligned}$$

where $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function, K is the number of negative samples v_n sampled from the "noisy distribution" of $p_n(v) = d_v^{3/4}$ as in [Mikolov *et al.*, 2013], and d_v is the output degree. Thus the partial derivative of Eq.(6) w.r.t. $\overrightarrow{u_i}$ can be rewritten as:

$$\begin{aligned} \frac{\partial O}{\partial \overrightarrow{u_i}^X} &= w_{ij}^X * \{[1-\sigma(\overrightarrow{u_j}'^{X^T}\cdot\overrightarrow{u_i}^X)]\overrightarrow{u_j}'^X \\ &-\sigma(\overrightarrow{u_n}'^{X^T}\cdot\overrightarrow{u_i}^X)\overrightarrow{u_n}'^X\} + \sum_{v_k\in X} p_a(v_i^Y|v_k^X) * w_{ij}^X \quad (9) \\ &*\{[1-\sigma(\overrightarrow{u_j}'^{Y^T}\cdot\overrightarrow{u_k}^X)]\overrightarrow{u_j}'^Y - \sigma(\overrightarrow{u_n}'^{Y^T}\cdot\overrightarrow{u_k}^X)\overrightarrow{u_n}'^Y\}\end{aligned}$$

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

$$\begin{aligned} \frac{\partial O}{\partial \overrightarrow{u_j}^X} &= w_{ij}^X * \{[1-\sigma(\overrightarrow{u_i}''^{X^T}\cdot\overrightarrow{u_j}^X)]\overrightarrow{u_i}''^X \\ &-\sigma(\overrightarrow{u_n}''^{X^T}\cdot\overrightarrow{u_j}^X)\overrightarrow{u_n}''^X\} + \sum_{v_k\in X} p_a(v_j^Y|v_k^X) * w_{ij}^X \quad (10) \\ &*\{[1-\sigma(\overrightarrow{u_i}''^{Y^T}\cdot\overrightarrow{u_k}^X)]\overrightarrow{u_i}''^Y - \sigma(\overrightarrow{u_n}''^{Y^T}\cdot\overrightarrow{u_k}^X)\overrightarrow{u_n}''^Y\}\end{aligned}$$

$$\frac{\partial O}{\partial \overrightarrow{u_i}''^X} = w_{ij}^X * [1 - \sigma(\overrightarrow{u_i}''^{X^T} \cdot \overrightarrow{u_j}^X)]\overrightarrow{u_j^X} + \sum_{v_k \in Y} p_a(v_j^X|v_k^Y) * w_{ij}^Y * [1 - \sigma(\overrightarrow{u_i}^{X^T} \cdot \overrightarrow{u_k}^Y)]\overrightarrow{u_k}^Y \tag{11}$$

$$\frac{\partial O}{\partial \overrightarrow{u_j}'^X} = w_{ij}^X * [1 - \sigma(\overrightarrow{u_j}'^{X^T} \cdot \overrightarrow{u_i}^X)]\overrightarrow{u_i}^X + \sum_{v_k \in Y} p_a(v_i^X|v_k^Y) * w_{ij}^Y * [1 - \sigma(\overrightarrow{u_j}'^{X^T} \cdot \overrightarrow{u_k}^Y)]\overrightarrow{u_k}^Y \tag{12}$$

$$\frac{\partial O}{\partial \overrightarrow{u_n}'^X} = w_{ij}^X * [-\sigma(\overrightarrow{u_n}'^{X^T} \cdot \overrightarrow{u_i}^X)\overrightarrow{u_i}^X] + \sum_{v_k \in Y} p_a(v_i^X|v_k^Y) * w_{ij}^Y * [-\sigma(\overrightarrow{u_n}'^{X^T} \cdot \overrightarrow{u_k}^Y)\overrightarrow{u_k}^Y] \tag{13}$$

$$\frac{\partial O}{\partial \overrightarrow{u_n}''^X} = w_{ij}^X * [-\sigma(\overrightarrow{u_n}''^{X^T} \cdot \overrightarrow{u_j}^X)\overrightarrow{u_j}^X] + \sum_{v_k \in Y} p_a(v_i^X|v_k^Y) * w_{ij}^Y * [-\sigma(\overrightarrow{u_n}''^{X^T} \cdot \overrightarrow{u_k}^Y)\overrightarrow{u_k}^Y] \tag{14}$$

With reference to Eqs.(9-14), the updating rules for network Y can be obtained by swapping the superscripts X and Y. They are not listed due to the page limit. The overall algorithm is shown in Algorithm 1:

Algorithm 1 Learning Aligned IONE Across Networks

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

Ensure: Node representations $\Theta = \{(\overrightarrow{u_x}^X, \overrightarrow{u_x}'^X, \overrightarrow{u_x}''^X), (\overrightarrow{u_y}^Y, \overrightarrow{u_y}'^Y, \overrightarrow{u_y}''^Y)\}$

procedure LEARNING(G^X,G^Y, E_a, η, K)
  Initialize $\Theta = \{(\overrightarrow{u_x}^X, \overrightarrow{u_x}'^X, \overrightarrow{u_x}''^X), (\overrightarrow{u_y}^Y, \overrightarrow{u_y}'^Y, \overrightarrow{u_y}''^Y)\}$
  repeat
    for N in $\{X, Y\}$ **do**
      Sample one edge (v_i,v_j) from G^N
      Update $\overrightarrow{u_j}'$, $\overrightarrow{u_i}$, $\overrightarrow{u_i}''$, $\overrightarrow{u_j}$ based on Eqs.(9-12) with η
      for $i = 0; i < K; i = i + 1$ **do**
        Sample a negative node v_n
        Update $\overrightarrow{u_i}$, $\overrightarrow{u_j}$, $\overrightarrow{u_n}'$, $\overrightarrow{u_n}''$ based on Eqs.(9, 10, 13, 14) with η
      end for
    end for
  until convergence
  return Θ
end procedure

3.4 Time Complexity

Sampling an edge takes constant time $O(1)$. Optimization using K negative samples takes $O(d(K+1))$ time, where d is the dimension. Therefore, the overall complexity for each step is $O(dK)$. In practice, the number of steps need for the optimization is usually proportional to the number of edges $|E|$ [Tang *et al.*, 2015b]. 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 $|V|$.

Table 1: Statistics of The Datasets Used for Evaluation

Networks	#Users	#Relations	#Anchors
Twitter	5,220	164,919	1,609
Foursquare	5,315	76,972	

3.5 Mapping Users Across Social Networks

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

$$rel(v_i^X, v_j^Y) = \frac{\sum_{p=1}^{d} u_{ip}^X \times u_{jp}^Y}{\sqrt{\sum_{p=1}^{d} {u_{ip}^X}^2} \times \sqrt{\sum_{p=1}^{d} {u_{jp}^Y}^2}} \tag{15}$$

So, for each user v_i^X in network X, we can find the most relevant user v_j^Y in network Y to be an anchor candidate.

4 Experiments

For performance evaluation, we employ two real-world social network datasets collected from Foursquare and Twitter [Zhang and Yu, 2015a]. Table 1 lists their statistics. The ground truth of anchors are provided in Foursquare profiles.

4.1 Comparative Methods

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

- **MAG**: A graph-based manifold alignment [Tan *et al.*, 2014] where the similarity of a linked user pair is defined as 1.
- **MAH**: A hypergraph-based manifold alignment [Tan *et al.*, 2014] where the hyperedges model the high-order relations in a social network.
- **CLF**: A method proposed in [Zhang and Yu, 2015a] 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.

Also, we abbreviate the proposed model with only the hard constraints on anchor nodes as *IONE* and that with also the soft constraints as illustrated in Sec.3.2 as *IONE-S*. And, we abbreviate the proposed model with only input context as *INE* and with only output context as *ONE*.

4.2 Evaluation Metrics

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

$$Precision@N = \frac{|CorrUser@N|^X + |CorrUser@N|^Y}{|UnMappedAnchors| \times 2} \tag{16}$$

Table 2: Performance Comparison - P@30

% Improvement	CRW (0.16)	MAH (0.32)	MAG (0.34)
INE (0.43)	168.75%	34.37%	26.47%
ONE (0.44)	175.00%	37.50%	29.41%
IONE (0.58)	262.50%	81.25%	70.58%

where $|CorrUser@N|$ is the number of unmapped anchor users with their corresponding users found among the top-N neighbors in the embedded space. $|UnMappedAnchors|$ is the total number of all unmapped anchor users.

Also, since the network alignment performance in general depends on the degree of overlapping of the two networks, we measure the degree as in [Tan *et al.*, 2014], given as

$$Interop(X,Y) = \frac{|Correlations| \times 2}{|Relations^X| + |Relations^Y|} \tag{17}$$

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

4.3 Experimental Results

The experimental results are presented in Figure 2. *IONE* is found to outperform all the baselines consistently and significantly given different @N settings as well as different training-to-test ratios. *MAH* and *MAG* perform better than *CRW*, showing that the random walk approach is not as accurate as the manifold alignment approach. However, both *MAG* and *MAH* fail to differentiate the follower-ship and followee-ship when constructing the incidence matrices of the hypergraph. We found that even *INE* and *ONE* can give better results, while *IONE* performs the best. Table2 tabulates our detailed quantitative improvement over the comparative baselines for P@30. Note that, different from our proposed method, the methods based on random walks (e.g.,*CLF* and *CRW*) are "asymmetric" in which taking a 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. For different training-to-test ratios, as observed from Fig.2(b), *IONE* outperforms all the baselines. Even for ratio settings as low as 10% to 20%, the performance enhancement is still significant.

Among the subspace learning based methods, we also compare their performance under the settings using representations of different dimensions. According to Fig.2(c), both *MAG* and *MAH* achieve good performance when the dimensionality setting is around 950, while *IONE* reaches good 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. We conclude that the proposed network embedding approach is significantly more effective and efficient than the matrix factorization-based approach.

Fig.2(d) shows how different methods perform at different values of *Interop*[1]. Intuitively, it will be easier to align the networks if they share more common edges. We observe that the performance of all the methods consistently achieve better results as the Interop value increases. Again, *IONE* consistently outperforms all the baseline methods, even when the resemblance of the networks is rather low.

For *IONE-S*, the soft constraint $p_a(v_i^X|v_k^Y)$ was derived similar to what being proposed in *CLF* [Zhang and Yu, 2015a]. By considering the labeled anchor links as positive data and those randomly sampled from unlabeled user pairs across networks as negative data, we compute structural features including common neighbors, extended Jaccard's coefficient and extended Adamic/Adar Measure for a SVM-Platt scaling classification model to estimate $p_a(v_i^X|v_k^Y)$. With the estimated $p_a(v_i^X|v_k^Y)$ incorporated as the soft constraints, the network embedding is then obtained via *IONE-S*. In general, the classification model achieves better performance when the training sets are more balanced. Here we use imbalance ratio $\frac{|-ve\ anchor\ links|}{|+ve\ anchor\ links|}$ as a proxy to reflect the performance of the empirical classification model. Fig.3(a) shows that *IONE-S* outperforms *CLF* where a similar SVM classification method is used. The performance of *CLF* is slightly better than that of the classification model indicating its high dependence on the ability of the classification model. For *IONE-S*, we found that it is very robust even when the accuracy of the estimated value of $p_a(v_i^X|v_k^Y)$ is limited. As there is an edge sampling process in learning *IONE* (See in Algorithm 1), whether the learning algorithm can converge is an important issue. Fig. 3(b) shows that *IONE-S* and *IONE* both converge in a stable manner. Specifically, *IONE-S* achieves its convergence much earlier than *IONE*. We believe the gain comes from the empirical soft constraints.

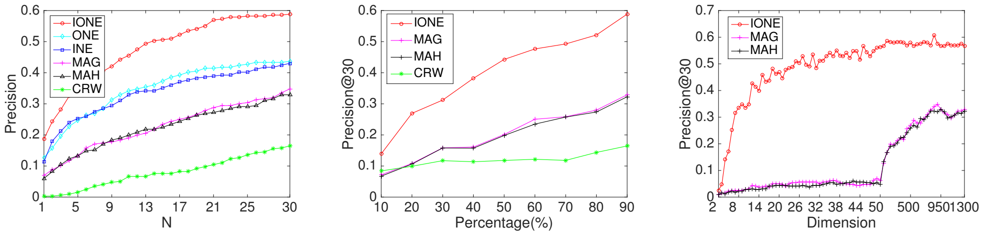
4.4 A Case Study

Here we plot Fig.4 to illustrate the effectiveness of *IONE* by showing two subgraphs, which are part of the corresponding social networks, and their embeddings in the inferred subspace. We adopt t-SNE [Van der Maaten and Hinton, 2008] for visualizating 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

1) ***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).

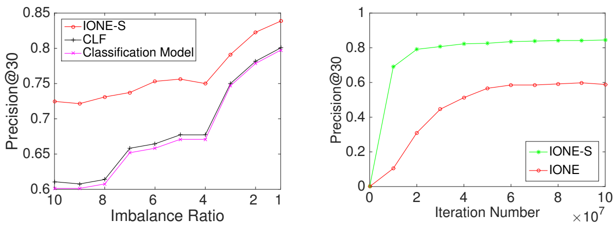
2) ***The anchor links do help the network alignment***. We observe that jam_fs has 3 input edges and 3 output edges from 3 anchor users *(kyl_an, hue_an, mil_an)* in the Foursquare network, while jam_tw also has 3 input and 2 output edges from the same set of anchor users in the Twitter network. In the plot of the embedded space (near the lower

[1] In our experiments, we vary the Interop value by removing non-anchor relations in the networks.



(a) $P@N$ (b) $P@30$ vs. Training ratio (c) $P@30$ vs. Dimension (d) $P@30$ vs. Interop

Figure 2: Detailed Performance Comparison on Twitter-Foursquare Dataset



(a) P@30 vs. Imbalance Ratio (b) P@30 vs. # of Iterations

Figure 3: Performance Comparison for IONE-S

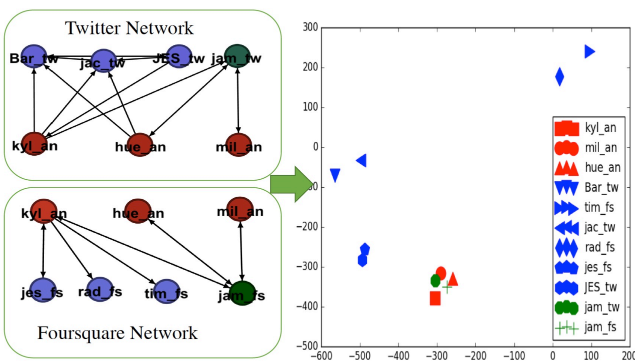


Figure 4: Subgraphs of Twitter-Foursquare Projected in Embedded Space.

side), jam_tw (green octagon) and jam_fs (green cross) are located very close to each other.

3) ***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).

5 Conclusion

We studied the problem of mapping users across networks. A representation learning model with the 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 ouput contexts. 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. The extensive experiments conducted on two real-world datasets demonstrate that our proposed model outperforms several state-of-the-art methods. Future work includes extending it to multiple networks and exploring its applicability to other types of network data.

Appendix I

We prove that our proposed formulation is equivalent to setting hard constraints to force the representations of the mapped anchor users equivalent. Suppose (v_i^X, v_i^Y) is an anchor link, and thus $p_a(v_i^X|v_i^Y) = p_a(v_i^Y|v_i^X) = 1$

$$\begin{aligned}\frac{\partial O}{\partial \overrightarrow{u_i}^X} &= w_{ij}^X\{[1-\sigma(\overrightarrow{u_j}'^{XT}\cdot\overrightarrow{u_i}^X)]\overrightarrow{u_j}'^X \\ &- \sigma(\overrightarrow{u_n}'^{XT}\cdot\overrightarrow{u_i}^X)\overrightarrow{u_n}'^X + [1-\sigma(\overrightarrow{u_j}'^{YT}\cdot\overrightarrow{u_i}^X)]\overrightarrow{u_j}'^Y \\ &- \sigma(\overrightarrow{u_n}'^{YT}\cdot\overrightarrow{u_i}^X)\overrightarrow{u_n}'^Y\}.\end{aligned} \tag{18}$$

The updating rule for $\overrightarrow{u_i}^Y$ can also be obtained:

$$\begin{aligned}\frac{\partial O}{\partial \overrightarrow{u_i}^Y} &= w_{ij}^X\{[1-\sigma(\overrightarrow{u_j}'^{YT}\cdot\overrightarrow{u_i}^Y)]\overrightarrow{u_j}'^Y \\ &- \sigma(\overrightarrow{u_n}'^{YT}\cdot\overrightarrow{u_i}^Y)\overrightarrow{u_n}'^Y + [1-\sigma(\overrightarrow{u_j}'^{XT}\cdot\overrightarrow{u_i}^Y)]\overrightarrow{u_j}'^X \\ &- \sigma(\overrightarrow{u_n}'^{XT}\cdot\overrightarrow{u_i}^Y)\overrightarrow{u_n}'^X\}.\end{aligned} \tag{19}$$

which is equivalent to Eq.(18) when we set the anchors' initial values as $\overrightarrow{u_i}^Y = \overrightarrow{u_i}^X$. This completes the proof.

6 Acknowledgement

This work has been partially supported by NSFC under Grant No. 61300178, National Program on Key Basic Research Project under Grant No. 2013CB329605, and Hong Kong Baptist University Strategic Development Fund.

References

[Dong *et al.*, 2012] Yuxiao Dong, Jie Tang, Sen Wu, Jilei Tian, Nitesh V Chawla, Jinghai Rao, and Huanhuan Cao. Link Prediction and Recommendation across Heterogeneous Social Networks. In *Proceedings of the IEEE 12th International Conference on Data Mining*, pages 181–190, Brussels, Belgium, December 2012. IEEE.

[Hu *et al.*, 2013] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. Personalized Recommendation via Cross-Domain Triadic Factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 595–606, Seoul, Korea, April 2013.

[Kong *et al.*, 2013] Xiangnan Kong, Jiawei Zhang, and Philip S Yu. Inferring Anchor Links across Multiple Heterogeneous Social Networks. In *Proceedings of the 22nd ACM international conference on information and knowledge management*, pages 179–188, San Francisco, CA, USA, October 2013. ACM.

[Koutra *et al.*, 2013] Danai Koutra, Hanghang Tong, and David Lubensky. BIG-ALIGN: Fast Bipartite Graph Alignment. In *Proceedings of IEEE 13th International Conference on Data Mining*, pages 389–398, Dallas, Texas, USA, December 2013. IEEE.

[Liu *et al.*, 2013] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen 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*, pages 495–504, Rome, Italy, February 2013. ACM.

[Liu *et al.*, 2014] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. HYDRA: Large-scale Social Identity Linkage via Heterogeneous Behavior Modeling. In *Proceedings of International Conference on Management of Data*, pages 51–62, Snowbird, UT, USA, June 2014. ACM.

[Malhotra *et al.*, 2012] Anshu Malhotra, Luam Totti, Wagner Meira Jr, Ponnurangam Kumaraguru, and Virgilio Almeida. Studying User Footprints in Different Online Social Networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining*, pages 1065–1070, Istanbul, Turkey, April 2012. IEEE Computer Society.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems 2013.*, pages 3111–3119, Lake Tahoe, Nevada, USA, December 2013.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710, NYC, USA, August 2014. ACM.

[Tan *et al.*, 2014] Shulong Tan, Ziyu Guan, Deng Cai, Xuzhen Qin, Jiajun Bu, and Chun Chen. Mapping Users across Networks by Manifold Alignment on Hypergraph. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Qubec City, Canada, July 2014.

[Tang *et al.*, 2015a] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive Text Embedding through Large-scale Heterogeneous Text Networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174. ACM, 2015.

[Tang *et al.*, 2015b] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, Florence, Italy, May 2015.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[Wu *et al.*, 2014] Shan-Hung Wu, Hao-Heng Chien, Kuan-Hua Lin, and Philip 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*, pages 298–306, Beijing, China, June 2014.

[Zafarani and Liu, 2013] Reza Zafarani and Huan 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*, pages 41–49, Chicago, IL, USA, August 2013. ACM.

[Zhang and Yu, 2015a] Jiawei Zhang and Philip S. Yu. Integrated Anchor and Social Link Predictions across Social Networks. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 2125–2132, Buenos Aires, Argentina, July 2015.

[Zhang and Yu, 2015b] Jiawei Zhang and Philip.S Yu. Multiple Anonymized Social Networks Alignment. In *Proceedings of IEEE 15th International Conference on Data Mining*, pages 389–398, Atlantic City, NJ, USA, November 2015. IEEE.

[Zhang and Yu, 2016] Jiawei Zhang and Philip S. Yu. PCT: Partial Co-Alignment of Social Networks. In *Proceedings of the 25th World Wide Web Conference*, WWW '16, 2016.

[Zhang *et al.*, 2014] Jiawei Zhang, Philip S Yu, and Zhi-Hua Zhou. Meta-path based Multi-Network Collective Link Prediction. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1286–1295, New York, USA, August 2014. ACM.

[Zhang *et al.*, 2015] Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, and Philip 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*, pages 1485–1494, Sydney, Australia, August 2015. ACM.