Learning Binary Hash Codes for Fast Anchor Link Retrieval across Networks

Yongqing Wang, Huawei Shen, Jinhua Gao and Xueqi Cheng
CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences
Beijing 100190, China
{wangyongqing,shenhuawei,gaojinhua,cxq}@ict.ac.cn

ABSTRACT

Users are usually involved in multiple social networks, without explicit *anchor links* that reveal the correspondence among different accounts of the same user across networks. Anchor link prediction aims to identify the hidden anchor links, which is a fundamental problem for user profiling, information cascading, and cross-domain recommendation. Although existing methods perform well in the accuracy of anchor link prediction, the pairwise search manners on inferring anchor links suffer from big challenge when being deployed in practical systems. To combat the challenges, in this paper we propose a novel embedding and matching architecture to directly learn binary hash code for each node. Hash codes offer us an efficient index to filter out the candidate node pairs for anchor link prediction. Extensive experiments on synthetic and real world large-scale datasets demonstrate that our proposed method has high time efficiency without loss of competitive prediction accuracy in anchor link prediction.

CCS CONCEPTS

• **Information systems → Information retrieval**; • **Computing methodologies → Machine learning**; • **Applied computing → Law, social and behavioral sciences**.

KEYWORDS

hash code, anchor link prediction, scalability

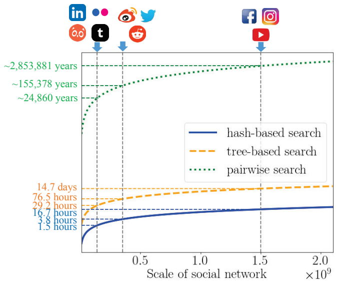
ACM Reference Format:
Yongqing Wang, Huawei Shen, Jinhua Gao and Xueqi Cheng. 2019. Learning Binary Hash Codes for Fast Anchor Link Retrieval across Networks. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313430

1 INTRODUCTION

With the benefit of socialized online services, users are often active across multiple online social networks simultaneously. The integration of user's data from multiple social networks can help to describe comprehensive user profile and interests so as to provide various of applications, including precision marketing and cyber security [3, 25, 29]. Meanwhile, the transferred user data from well established social media sites can mitigate the data sparsity and solve cold start problem for new founded sites [11, 15]. The key of bridging user identities across multiple social networks can be formalized as *anchor link prediction* (ALP) [1, 21, 22, 36], i.e., identifying hidden inter-network links connecting the accounts from the same users across different networks.

	#Visits across networks （Million）			
	f	👽	▶	🐦
f	-	-	748.8	78.7
👽	298.0	-	489.1	95.5
▶	1,030	292.1	-	165.2
🐦	19.4	-	44.3	-

(a) Traffics between the most popular social networks (from May 2018 to July 2018).



(b) Time consumption when executing 100 million ALP tasks on different scale of social networks by the different search strategies.

Figure 1: The necessity of anchor link prediction and high efficient solutions.

Most of existing studies in literatures attempt to improve inference accuracy leveraging types of information. The first category of methods relies on the profile information of users by pairwise similarity measures [12, 14, 33, 35]. However, the performance of these methods heavily depends on the availability and quality of user profile [4, 25, 30]. Consequently, these methods are difficult to be generalized to various scenarios [31, 32]. The second category of methods, in contrast, resorts to structural information, anticipating good generalization capability. Typical examples include exploiting local and global consistence pairwisely in topology and structures [7, 8, 19, 23, 24] across networks. Although existing methods perform rather well in the accuracy of anchor link prediction [9, 13, 16, 18, 20, 34], the pairwise search manners on inferring anchor links suffer from big challenge when being deployed in practical systems.

The most prominent challenge for anchor link prediction lies in *scalability*. According to the statistical report on the most popular social media[1], millions of co-visits exist between social media (see Figure 1(a)). However, the time consumption for searching anchor links would be extremely increased by the network size in traditional pairwise manners. As shown in the Figure 3(d), the

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313430

[1] https://www.statista.com/

time cost will be over 2 million years when executing 100 million ALP tasks on Facebook, Instagram or Youtube. Meanwhile, the time consumption is still up to 14.7 days even if applying tree-based search methods.

In this paper, we propose a novel architecture for learning binary hash codes across networks with characteristics in low storage cost and fast retrieval speed by the means of indexing candidate users in anchor link prediction. The goal of our proposed method is to map the structural features from different networks into a Hamming space of binary codes, preserving the similarity between user identities. The compact binary codes are capable of highly efficient retrieval on large-scale datasets [2, 5, 6, 27, 28]. Two anchor-link-aware stages, including embedding and matching, are introduced for efficient usage of labeled data in our proposed learning architecture. Firstly, we conduct network embedding on each network to capture structural features. Embedding method is compatible with learning effective representation of users in low-dimensional space. Besides, the learned embeddings are subject to the explicit constraints on labeled anchor links, eliminating the inference bias by inhomogeneous learning process across networks. In matching stage, the network embeddings from different networks are mapped into a common Hamming space which enables the retrieval process in an indexing way. More specifically, the main contributions of this work are three-folds:

- To solve the challenge in scalability, we propose a novel anchor link modeling method with learning binary hash codes according to network structures. The proposed modeling method is characteristic in low storage cost and fast retrieval speed.
- To solve the inhomogeneity in learned embeddings acr-oss networks, we introduce an anchor-link-aware constraint on network embedding, which can improve the prediction performance in the next matching stage.
- Experiments on time consumption show the potentials of our proposed method in real applications with large-scale networks. Meanwhile, the prediction performance resulted by our proposed method is also competitive to state-of-the-art methods.

2 PROBLEM DEFINITION

In this section, we introduce some basic notations and definitions in anchor link prediction task. A social network can be denoted as $G = \{\mathcal{U}, \mathcal{E}\}$, where $\mathcal{U}$ contains all user identities and $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{U}$ is the set of social relationships in the network. Without loss of generality, we focus on the anchor link prediction on two social networks, denoted as network G^s and network G^t respectively. Note that the settings of anchor link prediction on two networks can be easily extended to multiple social networks. For each user identity in one network, the objective of anchor link prediction is to identify, if any, its counterpart in another network. This task can be formally formulated as follows.

Definition 2.1. **Anchor link prediction:** Given two social networks $G^s = \{\mathcal{U}^s, \mathcal{E}^s\}$ and $G^t = \{\mathcal{U}^t, \mathcal{E}^t\}$, the objective of anchor link prediction is to learn the discriminative function $\mathcal{F}$: $\mathcal{U}^s \times \mathcal{U}^t \rightarrow \{0, 1\}$, which estimates whether a pair of user identities from two networks belong to the same nature person, i.e.,

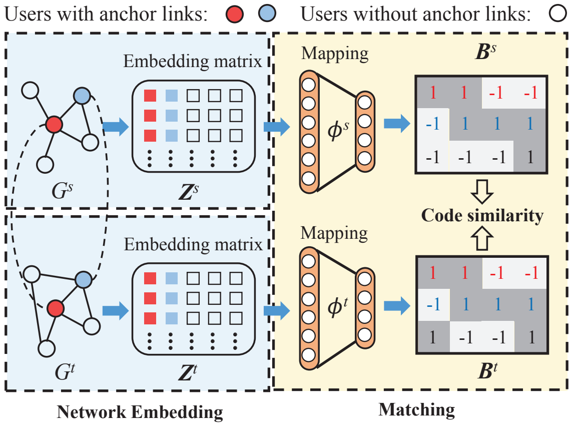


Figure 2: The framework of our proposed HALF model.

$$\mathcal{F}(u, v) = \begin{cases} 1, \text{if } u \in \mathcal{U}^s \text{ and } v \in \mathcal{U}^t \text{ belong to same person}, \\ 0, otherwise. \end{cases}$$

A labeled set of anchor links, denoted as $\mathcal{A} = \{(u, v)|u \in \mathcal{U}^s, v \in \mathcal{U}^t, \mathcal{F}(u, v) = 1\}$, is also provided as the supervised information for learning function $\mathcal{F}$.

We propose a supervised **H**ash-b**A**sed **L**earning **F**ramework for ALP with two anchor-link-aware stages, called HALF. The architecture of proposed model is shown in Figure 2. In this model, the first stage conducts network embedding on each network respectively. The objective of network embedding is to compress the high dimensional sparse network structure (i.e., adjacent matrix) into a low dimensional dense representation, preserving social structures of users. The second stage maps network embedding from each network into a common Hamming space and identifies the anchor links in the mapping space. Formally, the objective function of our proposed model can be presented as

$$\min_{\mathbf{Z}^s, \mathbf{Z}^t, \mathbf{B}^s, \mathbf{B}^t} \mathcal{L} = O\left(G^s, G^t, \mathcal{A}|\mathbf{Z}^s, \mathbf{Z}^t\right) + \mathcal{F}\left(\mathcal{A}|\mathbf{B}^s, \mathbf{B}^t\right), \qquad (1)$$

where $\mathbf{Z}^s$ and $\mathbf{Z}^t$ are the embedding matrices of network G^s and G^t respectively. The anchor-link-aware embedding loss $O\left(G^s, G^t, \mathcal{A}|\mathbf{Z}^s, \mathbf{Z}^t\right)$ reflects the capability of generating networks by $\mathbf{Z}^s$, $\mathbf{Z}^t$ under the constraints of $\mathcal{A}$. In the second part of the objective function, the symbol $\mathbf{B}$ refers to sign(Φ($\mathbf{Z}$)) with each element being defined as follows:

$$\mathbf{B} = \text{sign}(\Phi(\mathbf{Z})) = \begin{cases} 1, \Phi(\mathbf{Z}) \geq 0, \\ -1, \Phi(\mathbf{Z}) < 0, \end{cases}$$

where the mapping functions $\Phi : \mathbf{Z} \rightarrow \mathbf{R}^K$ try to map each user representation into a data point which can be easily transferred into a binary Hamming space. Note that the inputs and the parameters are independent in $\mathbf{B}^s$ and $\mathbf{B}^t$ respectively, we use the same notation for briefly illustrating the objective function. Therefore, the matching loss $\mathcal{F}\left(\mathcal{A}|\mathbf{B}^s, \mathbf{B}^t\right)$ depicts how well the learned binary hash codes capture the observed anchor links.

Direct optimization on Eq. (1) is quite difficult because of the interdependence between $\mathbf{Z}^s$, $\mathbf{Z}^t$, $\mathbf{B}^s$ and $\mathbf{B}^t$. Therefore, we turn to optimize the objective function by two stages separately, including network embedding and matching. Next we will introduce our proposed model in detail.

3 MODEL

In this section, we will introduce the detail of our proposed model by two anchor-link-aware stages: network embedding and matching.

3.1 Network embedding

Network embedding aims to embed a network into a low dimensional latent space, where each user is represented as a d-dimensional vector. The learned user representations are capable of depicting neighborhood similarity and community membership, which are important features for exploring anchor links. However, under the issues of anchor link prediction, independently learning embeddings for each network may lead to great matching errors when trying to map user representations from independent embedded spaces. Therefore, we introduce an anchor-link-aware network embedding method, exhibiting better performance in anchor link prediction.

Firstly, we introduce two vectors z_i and z_i' for user i, where z_i is user i's representation and z_i' is her *context* representation. For each directed edge $\langle u_i, u_j \rangle$, we define the probability of *context* u_j generated by user u_i as follows,

$$p\left(u_j | u_i\right) = \frac{\exp\left({z_j'}^T \cdot z_i\right)}{\sum_{k=1}^{|\mathcal{U}|} \exp\left({z_k'}^T \cdot z_i\right)}, \tag{2}$$

where $|\mathcal{U}|$ is the number of users in the network. The Eq. (2) captures the similarity between two users who share common neighborhoods. According to Eq. (2), the embedding loss of network G^s and G^t can be respectively formalized as

$$O\left(G^s\right) = -\sum_{\langle u_i^s, u_j^s \rangle \in \mathcal{E}^s} \log p\left(u_j^s | u_i^s\right), \tag{3}$$

and

$$O\left(G^t\right) = -\sum_{\langle u_i^t, u_j^t \rangle \in \mathcal{E}^t} \log p\left(u_j^t | u_i^t\right). \tag{4}$$

For circumventing the performance reduction introduced by independently embedding networks, we introduce constraints to all the nodes in the labeled anchor link set. For each anchor link $(u_i^t, u_j^s) \in \mathcal{A}$, we define the probability of its generation as

$$p\left(u_i^t, u_j^s\right) \sim \frac{z_i^t \cdot z_j^s}{\|z_i^t\| \|z_j^s\|}, \ \textit{iff}\ (u_i^t, u_j^s) \in \mathcal{A}, \tag{5}$$

and the negative logarithmic likelihood of anchor link set $\mathcal{A}$ can be formulated as

$$O\left(\mathcal{A} | G^s, G^t\right) = -\sum_{(u_i^t, u_k^s) \in \mathcal{A}} \log p\left(u_i^t | u_k^s\right) \tag{6}$$

According to Eq. (3), Eq. (4) and Eq. (6), we have the anchor-link-aware embedding loss in the embedding stage, described as

$$\min_{\mathbf{Z}^s, \mathbf{Z}^t} O\left(G^s\right) + O\left(G^t\right) + \gamma_e O\left(\mathcal{A} | G^s, G^t\right), \tag{7}$$

where γ_e is a hyper-parameter.

3.2 Matching

In previous works, finding one user's counterparts in other networks needs to loop over all possible candidates, resulting in huge, matching cost. In this paper, we adopt a learning method to map user representations from the original space into a Hamming space of binary codes. The binary codes can preserve the similarity in the original space and improve retrieval speed in anchor link prediction task. Thus, we introduce an end-to-end deep architecture for efficiently learning binary codes in the matching stage.

Firstly, we learn two mapping functions on each network, transferring the network embeddings into a continuous common space so as to determine whether the user identities are matched. We define $\phi_i^s = \Phi^s(z_i^s)$ and $\phi_j^t = \Phi^t(z_j^t)$, referring to the outputs of mapping function Φ^s and Φ^t with $z_i^s \in \mathbf{Z}^s$ and $z_j^t \in \mathbf{Z}^t$ serving as inputs respectively. The matching problem is formalized as a multi-class classification issue where the inputs are the outputs of mapping functions from two networks. We define the probability of the anchor link with u_i^s and its counterpart u_j^t in two networks as

$$p\left(u_j^t | u_i^s\right) = \frac{\exp\left({\phi_j^t}^T \cdot \phi_i^s\right)}{\sum_{u_k^t \in \mathcal{U}^t} \exp\left({\phi_k^t}^T \cdot \phi_i^s\right)}. \tag{8}$$

According to the probability defined in Eq. (8), the matching loss can be formulated as follows,

$$\mathcal{L}_{match} = -\sum_{(u_i^s, u_j^t) \in \mathcal{A}} \log p\left(u_j^t | u_i^s\right). \tag{9}$$

Then we learn the binary codes based on the continuous common space. We define the loss between binary codes and mapped user representations as

$$\mathcal{L}_{hash} = \|\mathbf{B}^s - \vec{\phi}^s\|_F^2 + \|\mathbf{B}^t - \vec{\phi}^t\|_F^2, \tag{10}$$

where $\vec{\phi}$ is the matrix of mapped user representations where the i-th column corresponds to user u_i. The loss limits the distance between the outputs of mapping functions and corresponding binary codes. The way of relaxation on binary codes avoids discrete learning problem, which may deteriorate the accuracy of the learned binary codes.

Overall, we define the loss of entire matching stage as follows,

$$\begin{aligned} \mathcal{F}\left(\mathcal{A} | \mathbf{B}^s, \mathbf{B}^t\right) = & \mathcal{L}_{match} + \gamma_m \mathcal{L}_{hash} \\ & + \eta_m \left(\|\vec{\phi}^s\|_F^2 + \|\vec{\phi}^t\|_F^2\right), \end{aligned} \tag{11}$$

where γ_m and η_m are hyper-parameters. The third term $\left(\|\vec{\phi}^s\|_F^2 + \|\vec{\phi}^t\|_F^2\right)$ regularizes the learned mapping results.

After obtaining the binary code of each user in two networks, we can use hashing-based search algorithms, e.g., locality sensitivity hashing, to calculate the similarity between binary codes and choose the most similar ones as the inferring results.

4 OPTIMIZATION

In this section, we develop algorithm to estimate the parameters of the proposed model.

4.1 Optimization on embedding stage

The direct optimization on Eq. (2) is computational expensive, requiring the summation over the entire set of users. To address this problem, we adopt the approach of negative sampling proposed in [17]. We define the negative sampling by the formalization as follows

$$\log \sigma \left({z'_j}^T \cdot z_i \right) + \sum_{k=1}^{K} E_{u_k \sim P_n(u)} \left[\log \sigma \left(-{z'_k}^T \cdot z_i \right) \right], \tag{12}$$

which is used to replace every $\log p \left(u_j | u_i \right)$ term in the objective function of the embedding stage. The function $\sigma(x) = 1/\left(1 + \exp(-x)\right)$ is the sigmoid function. The negative sampling distinguishs the edge $\langle u_i, u_j \rangle$ from K noise edges $\langle u_i, u_j \rangle$ sampled from noise distribution $P_n(u)$. We set the noise distribution $P_n(u) \propto {d_u}^{3/4}$, where d_u is the out-degree of user u. Besides, we introduce pre-trained process in order to stabilize the training process [10].

4.2 Optimization on matching stage

The optimization on logarithmic Eq. (8) also suffers high computational cost when summarizing compositions on all possible anchor links $\mathcal{A}$. To efficiently estimate the parameters on mapping, we maximize the $\log p \left(u_i^s, u_j^t \right)$ with negative sampling as

$$\log \sigma \left({\phi_j^t}^T \cdot \phi_i^s \right) + \sum_{k=1}^{K} E_{u_k \sim P'_n(u)} \left[\log \sigma \left(-{\phi_k^t}^T \cdot \phi_i^s \right) \right], \tag{13}$$

where the negative samples are drawn from $P'_n(u) = 1/|\mathcal{U}|$. Meanwhile, we construct a generalized linear function as the mapping functions with shared parameters across networks to handle the overfitting problem caused by scarce labeled anchor links in real applications, that is, $\phi_i = \tanh(w \cdot z_i + bias)$ where w is a parameter matrix.

According to the formalization on matching stage, the best matching performance is achieved when the binary codes from labeled anchor links are set to be the same. Hence we regularize $b_{(i,j)} = b_i^s = b_j^t$, *iff* $(u_i, u_j) \in \mathcal{A}$ in the training process. If we only consider the observed anchor links, the function $\|\mathbf{B}^s - \vec{\phi}^s\|_F^2 + \|\mathbf{B}^t - \vec{\phi}^t\|_F^2$ can be unfolded as

$$\sum_{(u_i, u_j) \in \mathcal{A}} \left(\|b_{(i,j)}\|^2 + \|\vec{\phi}_i^s\|^2 + \|\vec{\phi}_j^t\|^2 - 2b_{(i,j)}^T \cdot \left(\vec{\phi}_i^s + \vec{\phi}_j^t \right) \right).$$

It can be easily derived that the minimization of function $\|\mathbf{B}^s - \vec{\phi}^s\|_F^2 + \|\mathbf{B}^t - \vec{\phi}^t\|_F^2$ is equivalent to maximizing $b_{(i,j)}^T \cdot (\vec{\phi}_i^s + \vec{\phi}_j^t)$, when $\vec{\phi}^s$ and $\vec{\phi}^t$ are fixed. Therefore, we have $b_{(i,j)} = \mathrm{sign}(\vec{\phi}_i^s + \vec{\phi}_j^t)$.

At the end, we can reformulate the matching loss in Eq. (11) as follows,

$$\begin{aligned} \mathcal{F}'\left(\mathcal{A} | \mathbf{B}^s, \mathbf{B}^t \right) = & - \log \sigma \left({\phi_j^t}^T \cdot \phi_i^s \right) - \sum_{k=1}^{K} \left[\log \sigma \left(-{\phi_k^t}^T \cdot \phi_i^s \right) \right] \\ & + \gamma_m \sum_{(u_i, u_j) \in \mathcal{A}} \left(\|b_{(i,j)} - \vec{\phi}_i^s\|^2 + \|b_{(i,j)} - \vec{\phi}_j^t\|^2 \right) \\ & + \eta_m \left(\|\vec{\phi}^s\|_F^2 + \|\vec{\phi}^t\|_F^2 \right). \end{aligned}$$

The HALF algorithm for parameter estimation is described in Algorithm 1. Also we introduce optimization technics for achieving high computational efficiency and prediction accuracy, such as negative sampling and pre-training in embedding stage, parameter sharing in matching stage.

Algorithm 1 HALF

```
Input: Social network $G^s$ and $G^t$, labeled anchor links $\mathcal{A}$.
Output: Network embedding $\mathbf{Z}^s$ and $\mathbf{Z}^t$, binary hash code matrices $\mathbf{B}^s$ and $\mathbf{B}^t$.

Initialization: Initialize parameters with random values, including $\mathbf{Z}^s$, $\mathbf{Z}^t$, $\phi^s$ and $\phi^t$.
1. Embedding stage
while the value of objective function do not converge do
    Calculate $\partial O\left(G^s\right)/\partial \mathbf{Z}^s$ and $\partial O\left(G^s\right)/\partial \mathbf{Z}'^s$.
    Update $\mathbf{Z}^s$ and $\mathbf{Z}'^s$ with stochastic gradient descent.
end while
while the value of objective function do not converge do
    Calculate $\partial\left(O\left(G^t\right) + O\left(\mathcal{A}|G^s, G^t\right)\right)/\partial \mathbf{Z}^t$ and
    $\partial\left(O\left(G^t\right) + O\left(\mathcal{A}|G^s, G^t\right)\right)/\partial \mathbf{Z}'^t$ with fixed $\mathbf{Z}^s$.
    Update $\mathbf{Z}^t$ and $\mathbf{Z}'^t$ with stochastic gradient descent.
end while
2. Matching stage
while the value of objective function do not converge do
    Calculate $\partial \mathcal{F}\left(\mathcal{A}|\mathbf{B}^s, \mathbf{B}^t\right)/\partial \Phi^s$ and $\partial \mathcal{F}\left(\mathcal{A}|\mathbf{B}^s, \mathbf{B}^t\right)/\partial \Phi^t$.
    Update $\Phi^s$ and $\Phi^t$ with stochastic gradient descent.
end while
```

5 EXPERIMENT

In experiments, we compare our proposed method to the state-of-the-art methods for anchor link prediction on both synthetic data and real data. The results show that the proposed method achieves better performance in predicting anchor links with higher prediction accuracy and time efficiency.

5.1 Comparative methods

Previous anchor link prediction methods seldom can be well applied to large-scale data, suffering high computational cost in network embedding and matching. To better illustrate the performance of our proposed model, we choose scalable methods as baselines. The compared methods are summarized as follows:

- **FRUI-P** [37]: This work proposes an unsupervised method. We use the method to illustrate the performance when confronting with scarce label data.
- **MNA** [9]: This method models anchor link prediction as a classification problem based on human-craft features.
- **PALE-MLP** [16]: This method is a two-stage model, including separated network embedding and matching.
- **IONE** [13]: This method proposes a joint learning framework on embedding and matching.

As only orientation of vector is kept in Hamming space, hash codes would suffer from loss of information. For better illustrating the prediction accuracy, we introduce the codes from our proposed method without binarization to compare with the baselines.

5.2 Evaluation

We regard the prediction task of finding the matching node in another network as a ranking problem with the similarity score of two nodes' representations being served as ranking scores. Due to the high matching cost in baselines, 10 random nodes are sampled for each node. These sampled nodes, together with the ground truth matching node, are provided as candidate set. Each model is expected to correctly recognize the ground truth matching node from the candidate set. The prediction performance is evaluate by *Mean Reciprocal Rank* (MRR). Larger MRR values indicate better performance.

5.3 Experiments on synthetic data

The goal of experiments on synthetic data is to validate the effectiveness of our proposed models in prediction accuracy and time complexity.

5.3.1 Dataset. The synthetic data is built from a dataset crawled from Blogcatalog[2], processed by Tang and Liu [26]. The dataset contains 10,312 bloggers and 333,983 social links of bloggers. We follow the sampling strategy mentioned in [16] where $\alpha_s = 0.5$ and $\alpha_c = 0.8$ to sample two sub-networks from Blogcatalog network to serve as source network G^s and target network G^t in anchor link prediction. Every node in one sub-network has a matched node in the other sub-network, marked as anchor link. All labeled anchor links are randomly separated into two datasets by a fixed proportion, denoted as training and test datasets. Next, we conduct experiments with different settings on training data size in order to validate the effectiveness of our proposed model on label sparsity.

5.3.2 The effects of label sparsity. We examine the effects of training data size by varying the size from 5% to 60% of the entire labeled data. The experimental results are shown in Figure 3(a). In the figure, we can see that our proposed method achieves the best performance on MRR than all baselines under different settings of training data size. It is interested that FRUI-P is the second-best solution when merely given 5% of labels for training. It indicates that supervised methods would be invalid under the problem of label sparsity. However, our proposed method is still competitive to unsupervised method with scarce labeled data, showing the better performance with 5% of labeled data for training.

5.4 Experiments on real data

The goal of the experiments on real data is to demonstrate the effectiveness of our proposed models in real applications. The performance is evaluated by prediction accuracy and efficiency in retrieval time.

5.4.1 Dataset. Networks from two popular social networking services in China are adopted as datasets:

- **Douban**[3]: Douban offers services for their users to freely share favorite content through social network, including movies, books, music and other off-line events.
- **Sina Weibo**[4]: Weibo is a very popular microblogging platform in China. The number of registered users has exceeded 300 millions by the end of 2017.

The anchor link set is crawled though Douban data service API. In Douban, users would prefer to post the links to their own Weibo accounts in descriptions. According to this reliable information, we link 15,009 pairs of accounts across the two social networks. Then we extract the social networks relative to the labeled users in each network. The basic statistics of extracted real data is illustrated in Table 1. As described in the table, we can see that only 1.9% of total Weibo users and 0.7% of total Douban users have their anchor links being labeled in our dataset. The scarcely labeled anchor links with huge network size pose a big challenge in both optimization and matching in anchor link prediction. In the following experiments, we conduct two series of experiments, denoting "D → W" where we try to find the connected Weibo account for a Douban account and vise versa for "W → D".

Table 1: Real data description

	#Users	#Links	#Anchor links
Douban	2,046,509	6,493,150	15,009
Weibo	788,524	4,413,187	

5.4.2 The length of binary codes. The length of binary hash codes has great influence on captured structural properties, storage cost and retrieval speed. However, long binary codes may suffer overfitting problem in modeling. Therefore, we conduct experiments with code length varying in 24, 28, 32 and 36 bits respectively, to validate its influence on prediction performance. The 70% of entire labeled anchor links serves as training data. The experimental results are recorded in Figure 3(b). As we can see, the best prediction performance is achieved when the length of binary codes is 32 bits. Short bits of binary codes may cause more conflicts in matching, resulting in the reduction of prediction performance. However, long bits of binary codes may suffer overfitting problem when the labeled anchor links is not enough to learn the matching model. Thus, we set the length of binary codes to be 32 bits in our following experiments.

5.4.3 Evaluation on prediction accuracy. We conduct experiments on 70% of labeled data and examine prediction accuracy of all compared methods on the rest of labeled anchor links. As shown in the Figure 3(c), our proposed method significantly outperform all baselines on both prediction tasks "D → W" and "W → D". The MRR values from our proposed method are 0.32(±0.003) and 0.32(±0.001) on "D → W" and "W → D" respectively, achieving 8% and 10% improvement than the second-best solution.

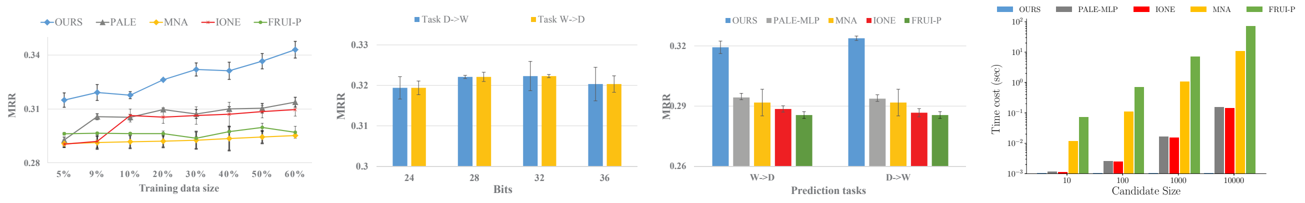
5.5 Time cost on inferring anchor links

With the binary hash codes, we can fast retrieve the potential anchor links by indexing way in our proposed method. The experiments are implemented with Python 2.7 on Intel(R) Xeon(R) CPU ES-2640 @ 2.40GHz with 10 cores and 128GB memory. We examine the time cost when inferring anchor links given 10, 100, 1,000 and 10,000 candidates. The experimental results are shown in Figure 3(d). We can see that our proposed method optimizes the inference time in order of magnitude comparing to other pair-wise solutions.

5.6 Other discussions

In this section, we discuss two remaining issues, i.e, effect of anchor-link-aware constraint formulated in Eq. (6) and the loss of binary codes with consideration of prediction accuracy. The experimental

[2] http://www.blogcatalog.com
[3] http://www.douban.com
[4] http://www.weibo.com



(a) (b) (c) (d)

Figure 3: (a) The experimental results on synthetic data with different training data size; (b) The experimental results on real data with different length of bits; (c) The prediction performance on real data; (d) Query time on 10, 100, 1,000 and 10,000 candidate size.

Table 2: The effect of anchor-link-aware constraint in embedding stage and the loss of binary codes in prediction accuracy.

	Synthetic data	Real data	
		W→D	D→W
Has Eq. (6)	**0.34(±0.005)**	**0.32(±0.003)**	**0.32(±0.001)**
No Eq. (6)	0.29(±0.003)	0.29(±0.005)	0.29(±0.004)
Cont. codes	**0.34(±0.005)**	**0.32(±0.003)**	**0.32(±0.001)**
Binary codes	0.29(±0.004)	0.29(±0.003)	0.29(±0.002)

p.s. Cont. codes refer to the codes without binarization.

results related to the two issues are show in Table 2. We can observe that the prediction performance is boosted with the supplementary of the anchor-link-awared constraint in embedding stage on both synthetic and real data. It means that the representation learning could be inductive to capture common structural features across networks, resulting high prediction accuracy. Besides, although the binary codes would cause loss of prediction accuracy in ALP, the results are still considerable when comparing to the MRR values resulted by PALE, MNA, IONE and FRUI-P methods under the same experimental settings which achieve to 0.31 ± 0.003, 0.29 ± 0.002, 0.31 ± 0.004, 0.30 ± 0.003 and 0.29 ± 0.002 on synthetic data, 0.29 ± 0.007, 0.28 ± 0.002 and 0.28 ± 0.002 on task “W→D”, and 0.29 ± 0.007, 0.29 ± 0.002, 0.28 ± 0.002 and 0.28 ± 0.002 on task “D→W” respectively.

6 CONCLUSION

In this paper, we point out one of big challenges for ALP, i.e., scalability when deployed in practical systems. To combat this problem, we propose a novel learning architecture with two stages for learning binary codes across networks. In embedding stage, we embed each user from high-dimensional and sparse raw representaion into a low-dimensional and dense representaion in each network. In matching stage, we propose an efficient matching model for learning binary codes based on learned user representations. The experimental results show the great potential of our proposed model in real applications.

7 ACKNOWLEDGEMENT

This work was funded by the National Natural Science Foundation of China under Grant Numbers 61802371, 61433014, 61425016 and the National Key Research and Development Program of China under grant number 2016QY03D0504. Hua-wei Shen is also funded by K.C Wong Education Foundation and the Youth Innovation Promotion Association of the Chinese Academy of Sciences.

REFERENCES

[1] Yi Cui, Jian Pei, Guanting Tang, Wo Shun Luk, Daxin Jiang, and Ming Hua. 2013. In Finding email correspondents in online social networks. *World Wide Web* 16, 2, 195–218.

[2] Bo Dai, Ruiqi Guo, Sanjiv Kumar, Niao He, and Le Song. 2017. Stochastic Generative Hashing. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. 913–922.

[3] Ruben Enikolopov, Maria Petrova, and Konstantin Sonin. 2018. Social media and corruption. *American Economic Journal: Applied Economics* 10, 1 (2018), 150–74.

[4] Oana Goga, Patrick Loiseau, Robin Sommer, Renata Teixeira, and Krishna P Gummadi. 2015. On the reliability of profile matching across large online social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1799–1808.

[5] Qing Yuan Jiang and Wu Jun Li. 2017. Deep cross-modal hashing. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*. 3270–3278. arXiv:1602.02255

[6] Qing-Yuan Jiang and Wu-Jun Li. 2018. Asymmetric Deep Supervised Hashing. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*.

[7] Gunnar W. Klau. 2009. A new graph-based method for pairwise global network alignment. *BMC Bioinformatics* 10, SUPPL. 1 (2009).

[8] Giorgos Kollias, Shahin Mohammadi, and Ananth Grama. 2012. Network Similarity Decomposition (NSD): A Fast and Scalable Approach to Network Alignment. *IEEE Transactions on Knowledge and Data Engineering* 24, 12 (2012), 2232–2243.

[9] Xiangnan Kong, Jiawei Zhang, and Philip S. Yu. 2013. Inferring anchor links across multiple heterogeneous social networks. In *Proceedings of the 22nd ACM International Conference on Conference on Information and Knowledge Management*. 179–188.

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521 (2015), 436–444.

[11] Chung-Yi Li and Shou-De Lin. 2014. Matching users and items across domains to improve the recommendation quality. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 801–810.

[12] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. 2013. WhatâĂŹs in a Name? An Unsupervised Approach to Link Users across Communities. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*. 495.

[13] Li Liu, William K Cheung, Xin Li, and Lejian Liao. 2016. Aligning Users Across Social Networks Using Network Embedding. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 1774–1780.

[14] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. 2014. HYDRA: Large-scale Social Identity Linkage via Heterogeneous Behavior Modeling. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA, 51–62.

[15] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: an embedding and mapping approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2464–2470.

[16] Tong Man, Huawei Shen, Shenghua Liu, Xiaolong Jin, and Xueqi Cheng. 2016. Predict anchor links across social networks via an embedding approach. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 1823–1829.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

[18] Xin Mu, Feida Zhu, Ee-Peng Lim, Jing Xiao, Jianzong Wang, and Zhi-Hua Zhou. 2016. User identity linkage by latent user space modelling. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1775–1784.

[19] Manikandan Narayanan and Richard M. Karp. 2007. Comparing Protein Interaction Networks via a Graph Match-and-Split Algorithm. *Journal of Computational Biology* 14, 7 (2007), 892–907.

[20] Yuanping Nie, Yan Jia, Shudong Li, Xiang Zhu, Aiping Li, and Bin Zhou. 2016. Identifying users across social networks based on dynamic core interests. *Neurocomputing* 210 (2016), 107–115.

[21] C Shi, Y Li, J Zhang, Y Sun, and P S Yu. 2017. A Survey of Heterogeneous Information Network Analysis. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2017), 17–37.

[22] Kai Shu, Suhang Wang, Jiliang Tang, Reza Zafarani, and Huan Liu. 2017. User Identity Linkage Across Online Social Networks: A Review. *ACM SIGKDD Explorations Newsletter* 18, 2 (2017), 5–17.

[23] Rohit Singh, Jinbo Xu, and Bonnie Berger. 2007. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Research in Computational Molecular Biology*. Springer, 16–31.

[24] Rohit Singh, Jinbo Xu, and Bonnie Berger. 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences* 105, 35 (2008), 12763–12768.

[25] Jiliang Tang, Yi Chang, and Huan Liu. 2014. Mining social media with social theories: a survey. *ACM SIGKDD Explorations Newsletter* 15, 2 (2014), 20–29.

[26] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining*. 817–825.

[27] Daixin Wang, Peng Cui, Mingdong Ou, and Wenwu Zhu. 2015. Deep multimodal hashing with orthogonal regularization. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 2291–2297.

[28] Dan Wang, Heyan Huang, Chi Lu, Bo-Si Feng, Liqiang Nie, Guihua Wen, and Xian-Ling Mao. 2018. Supervised Deep Hashing for Hierarchical Labeled Data. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*.

[29] Yongqing Wang, Huawei Shen, Shenghua Liu, Jinhua Gao, and Xueqi Cheng. 2017. Cascade dynamics modeling with attention-based recurrent neural network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2985–2991.

[30] Reza Zafarani and Huan Liu. 2009. Connecting Corresponding Identities across Communities.. In *Proceedings of the 3rd International Conference on Weblogs and Social Media*, Vol. 9. 354–357.

[31] Reza Zafarani and Huan Liu. 2014. Users Joining Multiple Sites : Distributions and Patterns User Membership Distribution across Sites. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. 635–638.

[32] Reza Zafarani and Huan Liu. 2016. Users joining multiple sites: Friendship and popularity variations across sites. *Information Fusion* 28 (2016), 83–89.

[33] Reza Zafarani, Lei Tang, and Huan Liu. 2015. User Identification Across Social Media. *ACM Transactions on Knowledge Discovery from Data* 10, 2 (2015), 16.

[34] Jiawei Zhang, Jianhui Chen, Shi Zhi, Yi Chang, S Yu Philip, and Jiawei Han. 2017. Link prediction across aligned networks with sparse and low rank matrix estimation. In *Proceedings of the 33rd IEEE International Conference on Data Engineering*. IEEE, 971–982.

[35] Jiawei Zhang, Xiangnan Kong, and Philip S. Yu. 2014. Transferring heterogeneous links across location-based social networks. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. 303–312.

[36] Jiawei Zhang and Philip S. Yu. 2015. Integrated anchor and social link predictions across social networks. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 2125–2132.

[37] Xiaoping Zhou, Xun Liang, Xiaoyong Du, and Jichao Zhao. 2017. Structure Based User Identification across Social Networks. *IEEE Transactions on Knowledge and Data Engineering* (2017).