

Deep Active Learning for Anchor User Prediction

Anfeng Cheng^{1,2}, Chuan Zhou^{1,2*}, Hong Yang³, Jia Wu⁴, Lei Li⁵, Jianlong Tan^{1,2}, Li Guo^{1,2}

¹Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

²School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

³School of Software, University of Technology Sydney, Sydney, Australia

⁴Department of Computing, Macquarie University, Sydney, Australia

⁵School of Computer and Information, Hefei University of Technology, Anhui, China
{chenganfeng, zhouchuan, tanjianlong, guoli}@iie.ac.cn, hong.yang@student.uts.edu.au
jia.wu@mq.edu.au, lilei@hfut.edu.cn

Abstract

Predicting pairs of anchor users plays an important role in the cross-network analysis. Due to the expensive costs of labeling anchor users for training prediction models, we consider in this paper the problem of minimizing the number of user pairs across multiple networks for labeling as to improve the accuracy of the prediction. To this end, we present a deep active learning model for anchor user prediction (DALAUP for short). However, active learning for anchor user sampling meets the challenges of non-i.i.d. user pair data caused by network structures and the correlation among anchor or non-anchor user pairs. To solve the challenges, DALAUP uses a couple of neural networks with shared-parameter to obtain the vector representations of user pairs, and ensembles three query strategies to select the most informative user pairs for labeling and model training. Experiments on real-world social network data demonstrate that DALAUP outperforms the state-of-the-art approaches.

1 Introduction

Social network users often join multiple social networks to obtain versatile services. For example, a user can be simultaneously active in both *Foursquare* and *Twitter*. These users are often termed as *anchor users* who often generate rich data for cross-network analysis. Advanced services for anchor users include cross-network recommendation [Li and Lin, 2014], link prediction [Zhang *et al.*, 2013] and information diffusion analysis [Peng *et al.*, 2013]. However, cross-network data generated from the anchor users are difficult to collect, because these users rarely use the same identities in different social networks, which poses difficulties for advancing cross network applications. Therefore, identifying anchor users across multiple social networks has attracted increasing research interests in recent years [Man *et al.*, 2016].

To identify anchor users, previous work [Liu *et al.*, 2013; Riederer *et al.*, 2016] try to collect and combine users' demo-

graphic data and daily-generated content data for estimation. For example, they often combine users' registration profile data such as names, genders and locations, and their daily-generated tweets, posts, blogs, reviews and ratings for analysis. However, it is difficult to obtain sufficient and correct demographic data and daily generated content data for prediction. As a result, the prediction accuracy of anchor users is often unstable and unsatisfactory.

Recently, researchers turn to social link data to predict anchor users [Liu *et al.*, 2016; Man *et al.*, 2016], because social link data, compared to demographic data and daily-generated content data, are much more reliable in terms of correctness and completeness. Existing works that use network structures for anchor user prediction can be divided into three categories, the unsupervised, supervised and semi-supervised models [Shu *et al.*, 2017], according to whether or to what extent the anchor user data can be observed and collected before model training.

The **unsupervised models** are proposed to solve unlabeled social structure data. These methods are proved to be equivalent to the network alignment problem [Singh *et al.*, 2007; Klau, 2009; Kollias *et al.*, 2012], which falls into the NP-hard combinatorial optimization. As a consequent, these approaches are either limited to small networks or only applicable to large but sparse networks [Man *et al.*, 2016]. The **supervised models** are used for predicting anchor users with training labels, which often generate accurate results when the number of labeled anchor users are adequate [Kong *et al.*, 2013; Man *et al.*, 2016]. Last but not least, the **semi-supervised models** can leverage both labeled and unlabeled training data for model training. Note that in semi-supervised models the unlabeled anchor users can be predicted during the learning process [Tan *et al.*, 2014; Liu *et al.*, 2016].

Obviously, the number of labeled anchor users plays a critical role in building an accurate anchor user prediction model. In order to reduce the number of training examples, Active Learning (AL) has been widely used to label training examples by iteratively selecting the most informative data for labeling at each round [Settles, 2009]. In this paper, we wish to design an economic anchor user prediction model based on the active learning method. To reduce the cost of labeling user pairs in social networks, we need to address the follow-

*Contact Author

ing challenges:

- **Challenge 1.** In this paper, active learning algorithm builds on *non-independently and identically distributed data*. Our anchor user prediction problem needs to consider social structure data obtained from multiple networks where both anchor and non-anchor user pairs exist.
- **Challenge 2.** The anchor user pairs and the non-anchor user pairs are interlocked (or correlated). The interlock means that if (v_A^i, v_B^m) is a pair of anchor users, $(v_A^i, v_B^{m'})$ and $(v_A^{i'}, v_B^m)$ are unlikely to be a pair of anchor users when $i \neq i'$ and $m \neq m'$. Hence, the query strategy of active learning should consider the interlock property among all the unlabeled user pairs.
- **Challenge 3.** There are two components in the anchor user prediction framework, i.e., the active sampling component and the anchor user prediction component. How to iteratively reinforce the two components and minimize the labeling cost is non-trivial.

To address the above challenges, we present a **Deep Active Learning model for Anchor User Prediction (DALAUP for short)**. The framework of the proposed method is illustrated in Fig. 1. From this figure, we can observe the three key components of the DALAUP method, i.e., the *anchor user prediction* component, the *query strategy* component and the *user pair selection* component. To solve **Challenge 1**, DALAUP uses a couple of convolution and deconvolution networks with shared-parameter to jointly analyze multiple social networks and obtain the representation vectors for cross-network user pairs, where structure information from both single-network and cross-network are jointly considered. To solve **Challenge 2**, in the query strategy component, DALAUP uses three different criteria for active learning query, i.e., the cross network structure aware information entropy, the cosine similarity and the expected error reduction. To solve **Challenge 3**, DALAUP adaptively selects user pairs by ensembling the three query methods to estimate the most informative user pairs for labeling. The main **contributions** of this work are summarized as follows,

- We are the first to study the problem of active learning for anchor user prediction across multiple social networks, where user data are non-i.i.d. distributed.
- We propose a new model DALAUP by integrating active sampling and anchor user prediction. Based on the rewards of the two components, they can reinforce each other via updating parameters iteratively.
- We conduct experiments on real-world data sets to verify the performance of DALAUP. The results demonstrate the effectiveness of our model compared with the state-of-the-art.

2 Preliminaries

Given two users in two different social networks, if they share the same identity in the real-world, we call them as an **anchor user pair**. The **anchor user prediction** aims to find out these anchor user pairs.

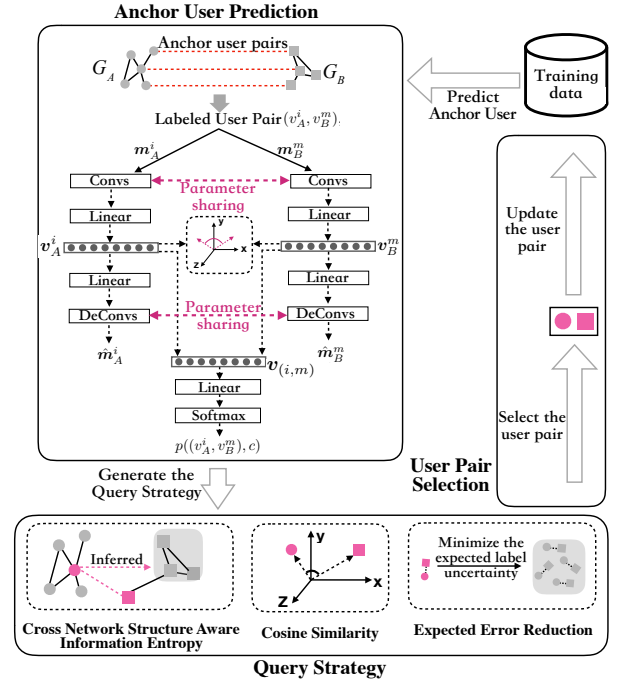


Figure 1: An illustration of the DALAUP model. DALAUP integrates three components for analysis, i.e., the anchor user prediction, the query strategy and the user pair selection. In the anchor user prediction component, vector representations of cross-network user pairs are estimated by a couple of neural networks with shared-parameters.

Formally, given two social networks represented as $\mathcal{G}_A = (\mathcal{V}_A, \mathcal{E}_A)$ and $\mathcal{G}_B = (\mathcal{V}_B, \mathcal{E}_B)$, where $\mathcal{V}_A, \mathcal{V}_B$ are two sets of users, and $\mathcal{E}_A = \{(v_A^i, v_A^j)\}$, $\mathcal{E}_B = \{(v_B^m, v_B^n)\}$ are two sets of undirected edges. Denote the sizes $N_A = |\mathcal{V}_A|$ and $N_B = |\mathcal{V}_B|$. Let $\mathcal{L} = \{(v_A^i, v_B^m)\}$ denote a labeled set of pairs of anchor (or non-anchor) users, where $v_A^i \in \mathcal{G}_A$ and $v_B^m \in \mathcal{G}_B$. Let $\mathcal{U} = \{\mathcal{V}_A \otimes \mathcal{V}_B - \mathcal{L}\}$ denote an unlabeled set, where \otimes is the Cartesian product. We wish to design an active anchor user prediction model which selects a small portion of pairs of users from \mathcal{U} . Then, based on the labeled pairs of users, we can build accurate prediction models for anchor users.

In the active learning, we aim to design a set of query functions $\{\phi_{qs}(v_A, v_B)\}$ and a selection function of pairs of users which can select a small portion of pairs of users $\{(v_A^*, v_B^*)\}$ from \mathcal{U} for labeling. This sampling and labeling process runs continuously until a given budget of labeling limit K is reached. In the next section, we will formally define the learning function.

3 The DALAUP Model

We introduce the DALAUP model for anchor user prediction under the active learning setting.

3.1 Anchor User Prediction

Given $\mathcal{G}_A, \mathcal{G}_B$ and \mathcal{L} , we first extract the *structural context* for each user in both networks \mathcal{G}_A and \mathcal{G}_B , based on which we

design a classifier for predicting the labels of pairs of users.

Structural Context. Given a social network \mathcal{G}_A , we wish to use a vector $p^{(i,s)} \in \mathbb{R}^{1 \times N_A}$ to represent each user $v_A^i \in \mathcal{G}_A$, where $p^{(i,s)}$ indicates the probability of visiting $v_A^j \in \mathcal{G}_A$ after s -step transitions initiated from v_A^i . The transitions are usually controlled by random walks with restart [Tong *et al.*, 2006] with a parameter $c \in (0, 1)$. Specifically, each transition either returns to v_A^i with probability c , or goes forward with probability $1 - c$. As a result, $p^{(i,s)}$ can be calculated iteratively as follows,

$$p^{(i,s)} = (1 - c) \cdot p^{(i,s-1)} \mathbf{D}_A + c \cdot p^{(i,0)}, \quad (1)$$

where \mathbf{D}_A is the normalized weighted matrix of \mathcal{G}_A with sums of the rows to be 1, and $p^{(i,0)}$ is the initial vector with the i -th entry to be 1 and the remaining entries 0.

To capture contextual information of network structures, we define the structural context \mathbf{m}_A^i of user v_A^i to be,

$$\mathbf{m}_A^i = \sum_{s=1}^S p^{(i,s)}, \quad (2)$$

where S indicates the number of steps. Following the same logic, we can obtain the structural context \mathbf{m}_B^m for each user $v_B^m \in \mathcal{V}_B$. These structural contexts are served as the input of the following neural network architecture.

Representation of Pairs of Users Across Networks. We wish to learn stable representations of pairs of users based on the structural context defined above. To preserve the structure information of a single network, we use both convolution and deconvolution neural networks to represent social networks [Niepert *et al.*, 2016]. The convolution neural network in our model consists of k convolution layers. Let $(\mathbf{x}_A^i)^{(0)} = \mathbf{m}_A^i \in \mathbb{R}^{1 \times N_A}$ be the input of user v_A^i at the first convolution layer and

$$(\mathbf{x}_A^i)^{(l)} = f(\text{Conv}(\mathbf{W}^{(l)}, (\mathbf{x}_A^i)^{(l-1)}) + b^{(l)}) \quad (3)$$

be the output of the l -th convolution layer, where $l = 1, 2, \dots, k$ and $\text{Conv}(\cdot)$ represents the convolution operation, $f(\cdot)$ is a non-linear activation function such as Sigmoid. $\mathbf{W}^{(l)}$ is a weight parameter and $b^{(l)}$ is the bias. We use $\mathbf{v}_A^i = h_{\theta^A}((\mathbf{x}_A^i)^{(k)})$ to denote the vector representation of user $v_A^i \in \mathcal{G}_A$, where $h_{\theta^A}(\cdot)$ is a fully connected linear layer with a parameter set $\theta^A = \{W_A, b_A\}$.

By using the deconvolution architecture, we can obtain the reconstruct representation $\hat{\mathbf{m}}_A^i$ for user v_A^i . Similarly, we can obtain representation \mathbf{v}_B^m and reconstruct representation $\hat{\mathbf{m}}_B^m$ for user $v_B^m \in \mathcal{G}_B$ with the parameters $\theta^B = \{W_B, b_B\}$ in full connected linear layer and the shared-parameters $(\mathbf{W}^{(l)}, b^{(l)})_{l=1}^k$ in convolution layers.

The goal of preserving the structure information of a single network is to minimize the reconstruction error as follows,

$$L_{\text{single}} = \frac{1}{|\mathcal{L}|} \sum_{(v_A^i, v_B^m) \in \mathcal{L}} (\|\mathbf{m}_A^i - \hat{\mathbf{m}}_A^i\|^2 + \|\mathbf{m}_B^m - \hat{\mathbf{m}}_B^m\|^2). \quad (4)$$

We consider anchor users to be similar in their representation spaces. Then, the loss of representing cross-network

information L_{cross} with respect to anchor users can be formulated as follows,

$$L_{\text{cross}} = \frac{1}{|\mathcal{L}^+|} \sum_{(v_A^i, v_B^m) \in \mathcal{L}^+} (1 - S(\mathbf{v}_A^i, \mathbf{v}_B^m)) \quad (5) \\ + \frac{1}{|\mathcal{L}^-|} \sum_{(v_A^i, v_B^m) \in \mathcal{L}^-} \max(0, S(\mathbf{v}_A^i, \mathbf{v}_B^m) - \epsilon),$$

where $S(\cdot)$ is a cosine similarity function, $\mathcal{L}^+ \subset \mathcal{L}$ is a set of pairs of anchor users, $\mathcal{L}^- \subset \mathcal{L}$ is a set of pairs of non-anchor users, and ϵ is the margin.

Anchor User Pair Classification. Based on the anchor user representation \mathbf{v}_A^i and \mathbf{v}_B^m discussed above, we further design a classifier to estimate a pair of users (v_A^i, v_B^m) , e.g., (v_A^i, v_B^m) is a pair of anchor users when $c = 1$, otherwise, a pair of non-anchor users. The probability of a pair of users (v_A^i, v_B^m) belongs to class $c \in \{0, 1\}$ can be defined as follows,

$$p((v_A^i, v_B^m), c) = \text{Softmax}(h_{\theta}(\mathbf{v}_{(i,m)})), \quad (6)$$

where $\mathbf{v}_{(i,m)} := (\mathbf{v}_A^i + \mathbf{v}_B^m)/2$ is an aggregated representation of a pair of users (v_A^i, v_B^m) , and $h_{\theta}(\cdot)$ is a fully connected linear layer with a parameter set $\theta = \{W, b\}$. The loss function under the cross entropy measure can be formulated as follows,

$$L_c = -\frac{1}{|\mathcal{L}|} \sum_{(v_A^i, v_B^m) \in \mathcal{L}} \left[y_{(v_A^i, v_B^m)} \log(p((v_A^i, v_B^m), 1)) \right. \\ \left. + (1 - y_{(v_A^i, v_B^m)}) \log(p((v_A^i, v_B^m), 0)) \right], \quad (7)$$

where $y_{(v_A^i, v_B^m)} \in \{0, 1\}$ is the true label of a pair of anchor users (v_A^i, v_B^m) .

In other words, the objective function of anchor user prediction can be formulated by minimizing the function as follows,

$$L_{\text{aup}} = L_{\text{single}} + \lambda_1 L_{\text{cross}} + \lambda_2 L_c + \lambda_3 L_{\text{reg}} \quad (8)$$

where $L_{\text{reg}} = \frac{1}{|\mathcal{L}|} \sum_{(v_A^i, v_B^m) \in \mathcal{L}} (\|\mathbf{v}_A^i\|^2 + \|\mathbf{v}_B^m\|^2)$ is an L_2 -norm regularizer term to prevent over-fitting and $\lambda_1, \lambda_2, \lambda_3$ are hyper-parameters to tradeoff the four parts.

3.2 Query Strategy

Based on the prediction of pairs of anchor users, we have two networks $\mathcal{G}_A, \mathcal{G}_B$, a labeled data set \mathcal{L} and an unlabeled data set \mathcal{U} . Moreover, we obtain the representations of all nodes in $\mathcal{G}_A, \mathcal{G}_B$, and the prediction probability $p((\cdot, \cdot), c)$ for unlabeled pair in \mathcal{U} . Then, we wish to design query strategies to choose the most informative user pairs for labeling.

The anchor user pairs and non-anchor user pairs are interlocked. Once an anchor user pair is labeled, we can infer a large set of non-anchor user pairs. Thus, anchor user pairs are more valuable and informative for labeling than non-anchor user pairs. Here, we design two strategies *cross network information entropy* and *cosine similarity* to find user pairs which are more likely to be anchor ones. Moreover, we propose the third query strategy *expected error reduction* to maximize the label certainty of the unlabeled use pairs.

Cross Network Structure Aware Information Entropy. Given two networks $\mathcal{G}_A, \mathcal{G}_B$ and a labeled set \mathcal{L} , according

to Eq. (6), we can obtain the probability $p((v_A^i, v_B^m), c)$ of the candidate user pair $(v_A^i, v_B^m) \in \mathcal{U}$ belonging to class c . The measure function for unknown label c of each unlabeled user pair (v_A^i, v_B^m) is defined as follows,

$$\phi_{saie}(v_A^i, v_B^m) = \sum_{c \in \{0,1\}} \left[p((v_A^i, v_B^m), c) \cdot R((v_A^i, v_B^m), c) \right], \quad (9)$$

where $R((v_A^i, v_B^m), c)$ is the reward function to measure the labeling outcome $((v_A^i, v_B^m), c)$. For information entropy, reward function $R((v_A^i, v_B^m), c)$ can be defined as follows,

$$R((v_A^i, v_B^m), c) = -\log p((v_A^i, v_B^m), c). \quad (10)$$

Due to the interlock between anchor user pairs and non-anchor user pairs, the reward for labeling anchor user pairs is much more important than labeling non-anchor ones. Let $\mathcal{S}_{(v_A^i, v_B^m)}$ denote a set of non-anchor user pairs inferred from (v_A^i, v_B^m) when it is labeled to be anchor one, we modify the reward function in Eq. (10) to be,

$$R((v_A^i, v_B^m), 0) = -\log p((v_A^i, v_B^m), 0)$$

and

$$R((v_A^i, v_B^m), 1) = -\log p((v_A^i, v_B^m), 1) - \sum_{s \in \mathcal{S}_{(v_A^i, v_B^m)}} \log p(s, 0). \quad (11)$$

Based on the new measure functions, we enable the cross network information entropy for candidate user pairs in \mathcal{U} , i.e., the larger $\phi_{ie}(v_A^i, v_B^m)$, the more informative (v_A^i, v_B^m) we obtain.

Cosine Similarity. After predicting the anchor users, we obtain the representations of v_A^i, v_B^m in the low-dimensional vector space. Since the anchor user pairs are similar between their representation vectors as in Eq. (5), we adopt the cosine similarity $cs(v_A^i, v_B^m)$ to query each candidate user pair $(v_A^i, v_B^m) \in \mathcal{U}$. The cosine similarity based query strategy ϕ_{cs} is defined as follows,

$$\phi_{cs}(v_A^i, v_B^m) = |cs(v_A^i, v_B^m)|. \quad (12)$$

The larger $\phi_{cs}(v_A^i, v_B^m)$ is, the more valuable (v_A^i, v_B^m) is.

Expected Error Reduction. For $(v_A^i, v_B^m) \in \mathcal{U}$, we introduce the expected error reduction query strategy, based on the prediction performance on the remaining unlabeled instances $\mathcal{U} \setminus (v_A^i, v_B^m)$ [Aggarwal *et al.*, 2014]. The expected error reduction aims to choose $(v_A^i, v_B^m) \in \mathcal{U}$ to maximize the label certainty of use pairs in $\mathcal{U} \setminus (v_A^i, v_B^m)$. Intuitively, the estimated label of $(\hat{v}_A^i, \hat{v}_B^m) \in \mathcal{U} \setminus (v_A^i, v_B^m)$ relates the probability $\hat{p}((\hat{v}_A^i, \hat{v}_B^m), c)$. Thus, the error reduction based query for (v_A^i, v_B^m) can be defined as follows,

$$\begin{aligned} \phi_{eer}(v_A^i, v_B^m) = & \sum_{c \in \{0,1\}} p((v_A^i, v_B^m), c) \cdot \\ & \left(\sum_{c \in \{0,1\}} \sum_{(\hat{v}_A^i, \hat{v}_B^m) \in \mathcal{U} \setminus (v_A^i, v_B^m)} |\hat{p}((\hat{v}_A^i, \hat{v}_B^m), c) - 0.5| \right), \end{aligned} \quad (13)$$

where $\hat{p}((\hat{v}_A^i, \hat{v}_B^m), c)$ is the probability that user pair $(\hat{v}_A^i, \hat{v}_B^m)$ belongs to class c , which can be calculated by Eq. (6) with $((v_A^i, v_B^m), c)$ being added to \mathcal{L} . The bigger $\phi_{eer}(v_A^i, v_B^m)$ is, the more helpful is (v_A^i, v_B^m) for maximizing the label certainty of pairs in $\mathcal{U} \setminus (v_A^i, v_B^m)$.

3.3 Active User Pair Selection Mechanism

Based on the above query strategies, a natural question is how to choose one candidate at each iteration e for labeling. As the prediction performance tends to get better when the labeled data set \mathcal{L} becomes larger after adding the labeled user pairs at each iteration, it is intuitive that the rewards are not independent random variables at different iterations. In this paper the reward $R_e(\phi)$ is defined as the average rise of Precision@30 and MAP@30, which can be estimated on the validate data. The recommended user pairs of the strategy are influenced by the recommendation results at previous iterations.

To simulate the adversarial setting, we use the multi-armed bandit method as the solution [Auer *et al.*, 2002]. To adaptively select the most informative user pairs, we adjust the ϵ -greedy algorithm to make a trade-off between exploitation and exploration, where the exploration probability ϵ is time-sensitive, e.g. $\epsilon \sim \text{Beta}(0.1, e)$ for different iteration $e \in \{1, 2, \dots, K/bs\}$. At the e -th iteration, we first sample an ϵ according to $\text{Beta}(0.1, e)$. The selection randomly selects a query strategy with probability ϵ (Exploration); Otherwise, it selects the one with the highest historical mean reward (Exploitation).

Specifically, let $Q_{e-1}(\phi)$ denote the historical mean reward of the query strategy ϕ at the first $e-1$ active learning iterations, then the query strategy ϕ_e at the e -th iteration can be chosen as below,

$$\phi_e = \begin{cases} \text{rand}(\phi_{saie}, \phi_{cs}, \phi_{eer}) & \text{with prob. } \epsilon \\ \arg \max_{\phi \in \{\phi_{saie}, \phi_{cs}, \phi_{eer}\}} Q_{e-1}(\phi) & \text{otherwise,} \end{cases} \quad (14)$$

where ϵ is a sample from $\text{Beta}(0.1, e)$. With the iteration time e increasing, ϵ is expected to decrease, which will reduce the attention onto the exploitation.

Let $n_{e-1}(\phi)$ be the time of query strategy ϕ employed at the first $e-1$ iteration. Assume the query strategy ϕ_{saie} is employed in the e -th iteration, then the historical mean reward and employ times of ϕ_{saie} can be updated through,

$$Q_e(\phi) := \frac{n_{e-1}(\phi) \cdot Q_{e-1}(\phi) + R_e(\phi)}{n_{e-1}(\phi) + 1} \quad (15)$$

and $n_e(\phi) := n_{e-1}(\phi) + 1$ with $\phi = \phi_{saie}$. Meanwhile, $Q_e(\phi_{cs}) := Q_{e-1}(\phi_{cs})$, $Q_e(\phi_{eer}) := Q_{e-1}(\phi_{eer})$, $n_e(\phi_{cs}) := n_{e-1}(\phi_{cs})$ and $n_e(\phi_{eer}) := n_{e-1}(\phi_{eer})$. The update is similar, when ϕ_{cs} or ϕ_{eer} is employed at the e -th iteration.

3.4 Algorithms

The algorithm for solving DALAUP is summarized in Algorithm 1, where Θ^e denotes a set of the learnable parameters in the *anchor user prediction* component at the e -th iteration, where $e \in \{0, 1, \dots, K/bs\}$.

Algorithm 1 The DALAUP algorithm.

Input:

Network \mathcal{G}_A and \mathcal{G}_B , labeled and unlabeled set $\{\mathcal{L}, \mathcal{U}\}$, maximum size K , size of small batch set bs , and query strategy set $\Phi := \{\phi_{saie}, \phi_{cs}, \phi_{eer}\}$.

Output:

Parameter set $\Theta^{K/bs}$ and ultimate prediction results.

```
1: Extract structural context  $\mathbf{m}_A, \mathbf{m}_B$ .
2: Learn initial parameter  $\Theta^0$  with initial  $\mathcal{L}$ .
3: Initialize  $n_0(\phi) \leftarrow 0, Q_0(\phi) = 0$  for each  $\phi \in \Phi$ .
4: Initialize  $e \leftarrow 1$ .
5: while  $e \leq K/bs$  do
6:    $\epsilon \sim \text{Beta}(0.1, e), \gamma \sim U[0, 1]$ 
7:   if  $\gamma < \epsilon$  then
8:      $\phi_e \leftarrow \text{rand}(\Phi)$ 
9:   else
10:     $\phi_e \leftarrow \arg \max_{\phi' \in \Phi} Q_{e-1}(\phi')$ 
11:   end if
12:   Estimate  $R_e(\phi_e)$  on validate data set.
13:   Update  $Q_e(\phi_e)$  with Eq. (15).
14:    $n_e(\phi_e) \leftarrow n_{e-1}(\phi_e) + 1$ .
15:    $n_e(\phi^*) \leftarrow n_{e-1}(\phi^*)$  for each  $\phi^* \in \Phi \setminus \phi_e$ .
16:    $Q_e(\phi^*) \leftarrow Q_{e-1}(\phi^*)$  for each  $\phi^* \in \Phi \setminus \phi_e$ .
17:   Choose  $bs$  candidate user pairs  $\mathcal{C} \subseteq \mathcal{U}$  according to
       query strategy  $\phi_e$  and label them.
18:    $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{C}, \mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{C}$ .
19:   Learn parameter  $\Theta^e$  with updated  $\mathcal{L}$ .
20:    $e \leftarrow e + 1$ .
21: end while
22: return  $\Theta^{K/bs}$  and ultimate prediction results.
```

4 Experiments

In this section, we compare DALAUP with existing baseline methods on real world social networks. Experimental results show the effectiveness of DALAUP compared with the state-of-the-art methods.

4.1 Data Sets

We use Foursquare and Twitter [Kong *et al.*, 2013] as the testbed. All the anchor user pairs are known in these data sets. The statistics are listed in Table 1.

Table 1: Statistics of Foursquare and Twitter

Network	User	Relation	Anchor User
Foursquare	5,313	76,972	
Twitter	5,120	164,919	3,141

We consider the labeled anchor user pairs as positive data and the inferred or labeled non-anchor user pairs as negative data. In the training process, we adopt undersampling to balance positive and negative data [Japkowicz and Stephen, 2002]. In our experiment, we randomly sample two non-anchor user pairs corresponding to an anchor user pair.

4.2 Baseline Methods and Evaluation Metrics

In order to evaluate the effectiveness of DALAUP, we choose the state-of-the-art unsupervised, supervised and

semi-supervised methods for comparison,

- **DW:** DeepWalk [Perozzi *et al.*, 2014]. An unsupervised method which first represents two social networks to a low-dimensional vector space and then calculates cosine similarity between vector representations of cross-network user pairs to predict anchor user.
- **PALE:** Predicting Anchor Links via Embedding [Man *et al.*, 2016]. A supervised model that learns network embedding with extended structural information.
- **IONE:** Input-output Network Embedding [Liu *et al.*, 2016]. A semi-supervised approach that gets aligned vector representations for multiple networks.

In addition, we introduce some variants of DALAUP for comparison, including

- **AUP:** Anchor User Prediction method proposed in this paper, which is a supervised manner.
- **DALAUP-saie, DALAUP-eer, DALAUP-cs, DALAUP-ie, and DALAUP-Rand.:** the variants of DALAUP distinguished by their different AL query strategies, i.e., DALAUP-saie is an active anchor user prediction model with only one query strategy ϕ_{saie} defined by Eq. (9) and Eq. (11). Same argument for DALAUP-eer and DALAUP-cs. Besides, DALAUP-ie is a simplified model of DALAUP-saie with a query strategy $\phi_{ie} := \phi_{saie}$ in which R is defined by Eq. (10). The last method DALAUP-Rand. is to randomly select user pairs from unlabeled set to label.

For each user in Foursquare, all above anchor user prediction algorithms output a list of candidate anchor users in Twitter. The Precision@30 [Liu *et al.*, 2016] and MAP@30 [Man *et al.*, 2016] are used as metrics for performance comparison.

4.3 Parameter Setting

The optimal parameter settings for each method are either determined by experiments or taken from the suggestions by previous works. Following [Perozzi *et al.*, 2014], we use the default parameter setting for DW, i.e., window size is 5 and walks per user is 80. In our method, we set the restart probability to $c = 0.6$ [Tong *et al.*, 2006] and the number of convolution layers $k = 2$. Margin ϵ in \mathcal{L}_{cross} is set to 0. The other parameters are set as: $\lambda_1 = 0.01, \lambda_2 = 0.01, \lambda_3 = 10^{-5}$.

4.4 Results and Analysis

To evaluate the effectiveness of DALAUP, we conduct two parts of experiments: effectiveness of AUP and effectiveness of DALAUP. For all experiments, we repeat the process 10 times and report the average results to test the statistical significance of the comparison results.

Effectiveness of AUP

In this experiment, different ratios of anchor user pairs are sampled randomly as positive data and the corresponding inferred non-anchor user as negative data for training. The remaining ones are viewed as test data. We set ratio $\eta \in \{0.1, 0.15, 0.2, \dots, 0.9\}$ respectively and fix the representation dimension $d = 56$ [Liu *et al.*, 2016]. Experiment results are shown in Fig. 2.

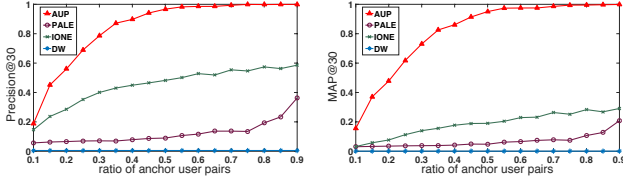


Figure 2: The performance comparisons of different anchor link prediction methods.

From Fig. 2, we can observe that the performance of our supervised method AUP is significantly better than the other comparing methods. Specially, when the ratio γ comes to 0.5, IONE can achieve about 48% and 19% in Precision@30 and MAP@30, while AUP can obtain 97% and 95% in these two measures.

Effectiveness of DALAUP

In this part, we investigate the effectiveness of the AL query strategies and the select mechanism proposed in section 3.2 and 3.3. To ensure that the performance difference in the experiments is solely caused by different AL query strategies and their select mechanism, we first randomly split the whole anchor user data into four parts: an initial training set to build an initial model, a validate set to get the rewards of different query strategies, a test set to evaluate the performance of the model, and an unlabeled set to select user pairs [Shen *et al.*, 2004].

The size of anchor user pairs in each part is 100, 300, 600 and 2,141 respectively. According to the interlock property, the non-anchor user pairs can be inferred from the anchor user pair set. At each iteration, we select a small batch of user pairs by following our AL query strategies, add the real label on them, and put them into the training set. The batch size $bs = 100$ and the maximum size $K = 1,500$. The performance comparisons of different query strategies with the different number of queried user pairs are shown in Fig. 3, in which the dotted line is the performance of the AUP model without any newly labeled information being added.

From Fig. 3, we can observe that:

- DALAUP significantly outperforms all the other methods in terms of both Precision@30 and MAP@30, which demonstrates that the proposed active user pair selection methods is effective in improving the prediction performance.
- The poor results of DALAUP-Rand. further validate the effectiveness of the proposed AL query strategies.
- Compared to information entropy DALAUP-ie, the modified DALAUP-saie method largely improves Precision@30 and MAP@30, which proves that cross network structure information can be used to ameliorate information entropy.

In addition, we investigate the performance of different anchor link prediction methods when adding 1500 extra labeled user pairs. These labeled data are selected in two manners. One is to select by DALAUP, while the other is to select ran-

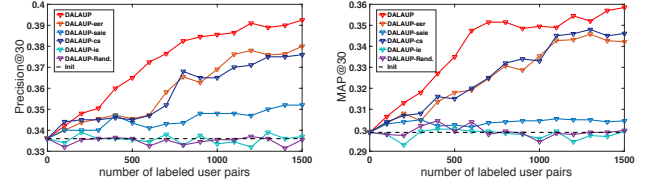


Figure 3: The comparisons of different query strategies.

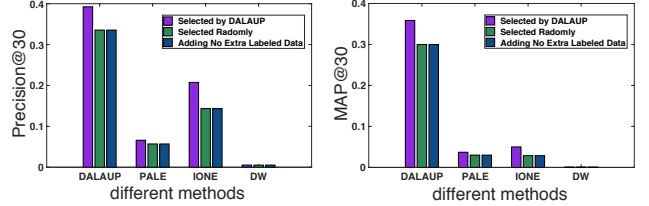


Figure 4: The comparisons of anchor link prediction methods when labeling 1500 user pairs.

domly. For the convenience comparison, we also add the performance of these methods with no extra labeled data added.

The results are shown in Fig. 4, from which we can observe that:

- DALAUP outperforms all the other methods, no matter the data selected by itself or by random.
- The labeled data, selected by DALAUP, can improve performance for the baselines.
- The randomly selected 1500 data for labeling cannot help better the benchmark methods, which confirms the importance of active selection methods.

5 Conclusion

In this paper, we present a deep active learning based method DALAUP to solve the anchor user prediction problem. We design a couple of convolution and deconvolution networks with shared-parameter to estimate the vector representations of cross-network anchor users. To solve the correlation difficulty of labeling anchor user pairs and non-anchor user pairs, we design three query strategies and a time-sensitive user pair selection algorithm for precisely selecting pairs of users for labeling. Experiments on real-world social network datasets demonstrate the effectiveness and efficiency of the proposed DALAUP method.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (No. 2016YFB0801301), the NSFC (No. 61872360), the MQNS (No. 9201701203), the MQEPS (No. 90275252 and No. 96804590), the MQRSG (No. 95109718), the Youth Innovation Promotion Association CAS (No. 2017210), and the Investigative Analytics Collaborative Research Project between Macquarie University and Data61 CSIRO.

References

- [Aggarwal *et al.*, 2014] Charu C Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S Yu. Active learning: A survey. 2014.
- [Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.
- [Japkowicz and Stephen, 2002] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.
- [Klau, 2009] Gunnar W Klau. A new graph-based method for pairwise global network alignment. *BMC bioinformatics*, 10(1):S59, 2009.
- [Kollias *et al.*, 2012] Giorgos Kollias, Shahin Mohammadi, and Ananth Grama. Network similarity decomposition (nsd): A fast and scalable approach to network alignment. *IEEE Transactions on Knowledge and Data Engineering*, 24(12):2232–2243, 2012.
- [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 & Knowledge Management*, pages 179–188. ACM, 2013.
- [Li and Lin, 2014] Chung-Yi Li and Shou-De Lin. 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*, pages 801–810. ACM, 2014.
- [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 ACM international conference on Web search and data mining*, pages 495–504. ACM, 2013.
- [Liu *et al.*, 2016] Li Liu, William K Cheung, Xin Li, and Lejian Liao. Aligning users across social networks using network embedding. In *IJCAI*, pages 1774–1780, 2016.
- [Man *et al.*, 2016] Tong Man, Huawei Shen, Shenghua Liu, Xiaolong Jin, and Xueqi Cheng. Predict anchor links across social networks via an embedding approach. In *IJCAI*, volume 16, pages 1823–1829, 2016.
- [Niepert *et al.*, 2016] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.
- [Peng *et al.*, 2013] Chuan Peng, Kuai Xu, Feng Wang, and Haiyan Wang. Predicting information diffusion initiated from multiple sources in online social networks. In *2013 Sixth International Symposium on Computational Intelligence and Design*, volume 2, pages 96–99. IEEE, 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. ACM, 2014.
- [Riederer *et al.*, 2016] Christopher Riederer, Yunsung Kim, Augustin Chaintreau, Nitish Korula, and Silvio Lattanzi. Linking users across domains with location data: Theory and validation. In *Proceedings of the 25th International Conference on World Wide Web*, pages 707–719. International World Wide Web Conferences Steering Committee, 2016.
- [Settles, 2009] Burr Settles. Active learning literature survey. *University of Wisconsinmadison*, 39(2):12–C131, 2009.
- [Shen *et al.*, 2004] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 589. Association for Computational Linguistics, 2004.
- [Shu *et al.*, 2017] Kai Shu, Suhang Wang, Jiliang Tang, Reza Zafarani, and Huan Liu. User identity linkage across online social networks: A review. *Acm Sigkdd Explorations Newsletter*, 18(2):5–17, 2017.
- [Singh *et al.*, 2007] Rohit Singh, Jinbo Xu, and Bonnie Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Annual International Conference on Research in Computational Molecular Biology*, pages 16–31. Springer, 2007.
- [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 *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
- [Tong *et al.*, 2006] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth International Conference on Data Mining (ICDM’06)*, pages 613–622. IEEE, 2006.
- [Zhang *et al.*, 2013] Jiawei Zhang, Xiangnan Kong, and S Yu Philip. Predicting social links for new users across aligned heterogeneous social networks. In *2013 IEEE 13th International Conference on Data Mining*, pages 1289–1294. IEEE, 2013.