Predict Anchor Links across Social Networks via an Embedding Approach

Tong Man,[12*] **Huawei Shen,**[1†] **Shenghua Liu,**[1†] **Xiaolong Jin,**[1†] and **Xueqi Cheng**[1†]
[1]{CAS Key Lab of Network Data Science and Technology
Institute of Computing Technology, Chinese Academy of Sciences, China}
[2]{University of Chinese Academy of Sciences, China}
*mantong@software.ict.ac.cn, †{shenhuawei, liushenghua, jinxiaolong, cxq}@ict.ac.cn

Abstract

Predicting anchor links across social networks has important implications to an array of applications, including cross-network information diffusion and cross-domain recommendation. One challenging problem is: *whether and to what extent* we can address the anchor link prediction problem, if only structural information of networks is available. Most existing methods, unsupervised or supervised, directly work on networks themselves rather than on their intrinsic structural regularities, and thus their effectiveness is sensitive to the high dimension and sparsity of networks. To offer a robust method, we propose a novel supervised model, called PALE, which employs network embedding with awareness of observed anchor links as supervised information to capture the major and specific structural regularities and further learns a stable cross-network mapping for predicting anchor links. Through extensive experiments on two realistic datasets, we demonstrate that PALE significantly outperforms the state-of-the-art methods.

1 Introduction

With the prosperity of online social networks, people often join multiple social networks [Zafarani and Liu, 2009; Sun *et al.*, 2012]. For example, one person could be an active user in both Facebook and Twitter. These shared users naturally form *anchor links* bridging different social networks [Kong *et al.*, 2013]. Anchor links are crucial to cross-network information diffusion [Peng *et al.*, 2013], link prediction [Ahmad *et al.*, 2010; Dong *et al.*, 2012], and cross-domain recommendation [Man *et al.*, 2015]. However, information about anchor links is often not available in practical scenarios, because most users have no motivation or are unwilling to explicitly correlate their identities in different online social networks [Backstrom *et al.*, 2007]. This poses the problem of *anchor link prediction*, i.e., identifying hidden anchor links across different social networks [Liu *et al.*, 2014; Tan *et al.*, 2014; Zhang *et al.*, 2015; Zhang and Yu, 2015]. Early studies address this problem either by leveraging self-reported user profiles (e.g., user name, profile picture, location, gender) and other demographical features [Iofciu *et al.*, 2011; Malhotra *et al.*, 2012] or by exploiting user generated contents, such as, tweets, posts, blogs, reviews, and ratings [Novak *et al.*, 2004; Liu *et al.*, 2013]. However, an important open problem is: *whether and to what extent* we can address the anchor link prediction problem, if only structural information of networks is available.

Existing approaches that leverage network structure for anchor link prediction fall into two main categories. The first category of approaches works in an *unsupervised* manner, where no information about explicit correspondence across networks is leveraged. These approaches cope with the anchor link prediction problem as a problem of network alignment, which is generally an NP-hard combinatorial optimization problem [Singh *et al.*, 2007; Klau, 2009; Kollias *et al.*, 2012], and solve it by finding certain structural similarity between nodes across networks. Consequently, these approaches either are limited to networks with moderate size, or can only be applicable to large scale networks under a sparse assumption [Bayati *et al.*, 2009]. The second category of approaches is *supervised* [Kong *et al.*, 2013; Zafarani and Liu, 2013], with which the anchor link prediction problem is solved under the supervision of observed anchor links (e.g., some users may explicitly mark their accounts in different social networks in their homepages, social media channels, or third-party websites like Aboutme.com). Most existing supervised approaches directly work on structural features of social network, such as, degree, clustering coefficient, the number of involved triangles, common neighbors, to name a few [Cui *et al.*, 2013; Kong *et al.*, 2013]. Without capturing intrinsic structural regularities of social networks, these approaches are particularly sensitive to network structure, and thus slight changes or noises of network structure may result in remarkably different results. In sum, we still lack an effective approach that can make the best of the structural regularities of social networks and the information pertaining to observed anchor links.

To bridge this gap, in this paper we propose a novel supervised model, called PALE (Predicting Anchor Links via Embedding), to tackle the anchor link prediction problem. This model contains two anchor-link-aware stages, namely, embedding and matching. Specifically, given two networks for predicting their anchor links, we first conduct network embedding on each network to capture its major structural regularity. Unlike existing methods that directly work on

human-defined structural features, we embed each network into a low-dimensional space to learn effective representation of nodes. In this way, major structural regularity of networks is well captured, while insignificant details are filtered away. Simultaneously, the network embedding preserves specific structural regularities, leveraging observed anchor links as supervised information. This method makes our model robust to slight changes of network structure. Next, in the matching stage, taking the low-dimensional representations of nodes as features, we learn a mapping function across the two learned low-dimensional spaces, supervised by observed anchor links. In order to offer the flexibility that the two latent spaces can be nonlinearly correlated, Multi-Layer Perceptron (MLP) is employed to learn the mapping function. Finally, for each node in one network, we identify the most likely counterpart in the other network according to the learned mapping function. Besides the robustness, another important merit of the proposed model is that the low-dimensional representation of network structure can be easily incorporated with contents and demographic features to further improve the accuracy of anchor link prediction.

We validate the proposed anchor link prediction model on two scenarios. The first one is to predict anchor links across two partially observed networks, which are sampled from the same social network. The second one intends to predict anchor links across two co-author networks in different research areas, namely Artificial Intelligence and Data Mining. Experimental results convincingly suggest that the proposed PALE model outperforms the compared baselines. Extensive analysis is also conducted to demonstrate the good performance of PALE under different settings of network structure.

2 Predict Anchor Links via Embedding

Denote a social network as $G = \{V, E\}$, where V is the set of nodes, and the edge set $E \subset (V \times V)$ reflects social relationships among nodes (e.g., friendships in Facebook). In this paper, we consider such a scenario: some users are simultaneously involved in two different social networks, forming anchor links across the two networks. Without loss of generality, we refer to one network as source network, and the other as target network, denoted with G^s and G^t respectively. For each node in the source network, we aim to identify, if any, its counterpart in the target network. This can be formally formulated as the following anchor link prediction problem.

Anchor link prediction: Given two networks $G^s = \{V^s, E^s\}$ and $G^t = \{V^t, E^t\}$ and a set of observed anchor links $T = \{(v, u)|v \in V^s, u \in V^t\}$, it aims to identify hidden anchor links across G^s and G^t.

To solve the anchor link prediction problem, we propose a novel supervised model, called PALE. In this model, the source and target networks are both embedded into low-dimensional spaces, denoted as Z^s and Z^t respectively, and a mapping function $\phi : Z^s \rightarrow Z^t$ across the two spaces is learned. Formally, PALE aims to find the optimal Z^s, Z^t, and ϕ by

$$\min_{Z^s, Z^t, \phi} \Big\{ L_e(G^s, Z^s, G^t, Z^t, T) + L_m(\phi, Z^s, Z^t, T) \Big\}, \quad (1)$$

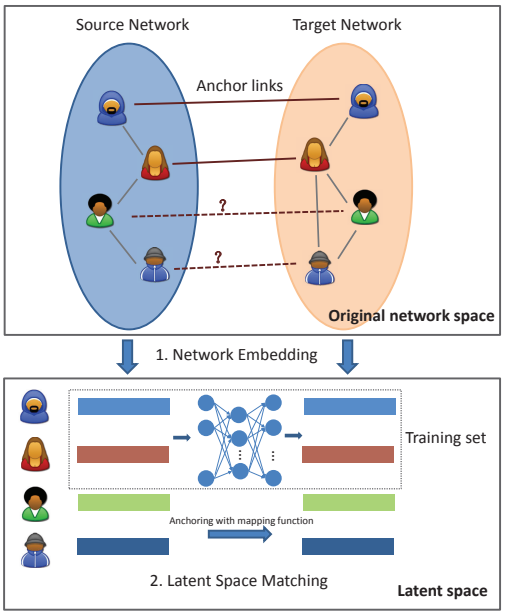


Figure 1: Illustrative diagram of the PALE model.

where $L_e(G^s, Z^s, G^t, Z^t, T)$ is the loss when embedding the source network G^s and the target network G^t into the low-dimensional spaces Z^s and Z^t, and the matching loss $L_m(\phi, Z^s, Z^t, T)$ reflects whether or not the observed anchor links in T are correctly predicted by the mapping function ϕ with Z^s and Z^t as its inputs.

Unfortunately, it is very difficult to solve this optimization problem efficiently because of the interdependence between Z^s, Z^t and ϕ. For this reason, we turn to find an approximately optimal solution, working in a two-stage embedding and matching manner, as shown in Figure 1.

2.1 Anchor-link-aware Network Embedding

In this stage, each network is embedded into a low-dimensional latent space, where each node v_i is represented as a d-dimensional vector z_i. A crucial problem for anchor link prediction is that some edges that exist in practice may be unobserved, as they have not been explicitly built or fail to be crawled. These missing edges can lead to unreliable representations when embedding networks into latent spaces. To combat this problem, we propose a strategy to identify hidden edges with the help of the observed anchor links and the structure of the other network. Based on the identified unobserved edges together with the observed ones, a reliable network embedding is learned.

Cross Network Extension

Before network embedding, we first leverage observed anchor links to extend both source and target network. It is usually true that given a pair of users with anchor links, if they have a connection in one network, so do their counterparts in the other network [Bayati *et al.*, 2009]. Based on such an observation, if two nodes are not linked in one network, but their counterparts are linked in the other network, we can add an edge between them in the present network, as shown in Fig-

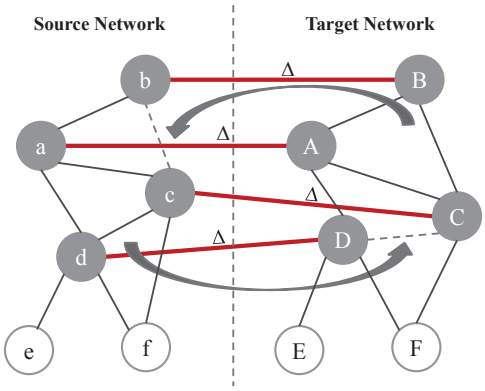


Figure 2: Network extension with the help of observed anchor links. The bold lines with symbol '△' (e.g., $a-A$ and $b-B$) represent anchor links. The dashed lines are the missing edges that are not observed in one network, but can be assumed as being present with the help of the counterpart edges in the other network. For example, the node pair (b, c) is not linked in the source network, but the counterpart node pair (B, C) is linked in the target network. This offers us a clue to extend the source network with edge $b-c$.

ure 2. Formally, given two networks G^s and G^t with anchor links T, the extended network $\tilde{G}^s$ of the source network G^s can be described as

$$\begin{aligned}\tilde{V}^s =& V^s \\ \tilde{E}^s =& E^s \cup \{(v_m^s, v_n^s) : (v_m^s, u_k^t) \in T, \\ & (v_n^s, u_l^t) \in T, (u_k^t, u_l^t) \in E^t\}.\end{aligned} \tag{2}$$

Similarly, the target network G^t is extended into $\tilde{G}^t$. Note that cross-network extension is not a mandatory requirement in our model.

Network Embedding

With the extended source and target networks, we embed them independently into latent spaces. For convenience, in this subsection, we use the same notations without distinguishing the source and the target network. For a pair of nodes v_i and v_j, given their d-dimensional representations z_i and z_j, the probability that an edge is observed between them is

$$p(v_i, v_j) = \sigma(z_i^T \cdot z_j) = \frac{1}{1 + e^{-z_i^T \cdot z_j}}, \tag{3}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function.

To learn the latent representation, we maximize the log-likelihood of the extended network $\tilde{G}$

$$\sum_{(i,j)\in\tilde{E}} \log p(v_i, v_j) = \sum_{(i,j)\in\tilde{E}} \log\sigma(z_i^T \cdot z_j). \tag{4}$$

Since only observed edges, including extended edges if any, are modeled, there exists a trivial solution when maximizing the objective function in Eq. (4): $z_{ik} = \infty$. To avoid the trivial solution, for each observed edge (v_i, v_j), we maximize the objective function with negative sampling as

$$\log\sigma(z_i^T \cdot z_j) + \sum_{k=1}^{K} E_{v_k \propto P_n(v)}[\log(1 - \sigma(z_i^T \cdot z_k))], \tag{5}$$

where the first term models the observed edge, the second expectation term samples negative edges from a null model where each node is sampled with the probability $P_n(v) \sim d_v^{3/4}$ as proposed in [Mikolov *et al.*, 2013b] and K is the number of sampled negative edges, d_v is the degree of node v.

Note that maximizing the summation of Eq. (5) over all the edges in the extended network $\tilde{G}^s$ and $\tilde{G}^t$ independently is an effective way to approximately minimize the loss of network embedding L_e in Eq. (1). Finally, we adopt stochastic gradient descent to learn the latent representations.

2.2 Supervised Latent Space Matching

Next, we turn to learn a mapping function ϕ, supervised by the observed anchor links $(v_l^s, u_n^t) \in T$ with latent representations being z_l^s and z_n^t.

Given $z_l^s \in Z^s$, the mapping function parameterized as $\phi(z_l^s; \Theta)$ projects it into the target space Z^t. Here, Θ is the collection of all parameters in the mapping function ϕ. The loss is

$$L_m(\phi, Z^s, Z^t, T) = \sum_{(v_l^s, u_n^t)\in T} ||\phi(z_l^s; \Theta) - z_n^t||_F, \tag{6}$$

where $||\cdot||_F$ is the Frobenius norm, which is a distance metric between the predicted latent factor and the training latent factor.

In this paper, we consider both linear and non-linear mapping functions. For the linear mapping function, Θ is a $d \times d$ matrix, as follows

$$\phi(z_l^s; \Theta) = \Theta \times z_l^s, \;\; v_l^s \in V^s. \tag{7}$$

Eq. (6) finds the best matrix Θ so that $\Theta \times z_l^s$ closely approximates z_n^t for all labelled pairs $(v_l^s, u_n^t) \in T$. Indeed, a linear mapping function has been successfully used in the previous study [Mikolov *et al.*, 2013a].

In addition, we employ Multi-Layer Perceptron (MLP) [Ruck *et al.*, 1990] to capture the non-linear relationship between the source and target spaces. In this way, the two spaces obtained in the network embedding stage are not required to be linearly aligned, which offers the embedding stage more flexibility to capture the structural regularities of networks.

2.3 Anchor Link Prediction

To make prediction, for any given node v_l^s in the source network with its representation z_l^s, we map it into the target latent space according to the mapping function $\phi(z_l^s; \Theta)$. We then predict hidden anchor links by identifying the counterpart node u_n^t that is the closest one to $\phi(z_l^s; \Theta)$ by

$$\min_n ||\phi(z_l^s; \Theta) - z_n^t||_F. \tag{8}$$

Alternatively, for each node in the source network, we can offer a list of nodes in the target network as its potential counterparts.

2.4 Complexity Analysis

In the network embedding stage, the total time complexity for one network G is $O(kd|E|)$, where k is the number of iterations, d is the dimension of the embedding vectors, $|E|$ is the

number of edges in the network. In the latent space matching stage, the two models with linear and non-linear mapping functions have different time complexity. For the case of the linear mapping function, the time complexity of solving each row of the transfer matrix as a sub-problem is $O(|T|d^2)$, thus the total complexity is $O(|T|d^3)$, where $|T|$ is the size of the set of observed edges; for the case with MLP as the non-linear mapping function, the time complexity is $O(kd|T|)$, where k is the iteration times, $|T|$ is the size of the set of observed edges. For prediction, the time complexity to match nodes across networks is $O(|V|^2)$, where $|V|$ is the number of nodes in the network.

3 Experiments

In this section, we compare the proposed PALE model with existing baseline methods on two datasets. One contains networks sampled from the Facebook network. The other consists of two co-author networks in different research areas, namely, Artificial Intelligence and Data Mining.

3.1 Methods and Metrics for Comparison

We choose both unsupervised and supervised methods for comparison, including

- Degree-Based Alignment: Nodes between two networks are matched in terms of their degrees, offering a trivial baseline for anchor link prediction.
- Matching Across Domains (MAD) [Li and Lin, 2014]: It matches shared nodes across homogenous networks through singular value decomposition, which is an unsupervised model for anchor link prediction.
- Multi-Network Anchoring (MNA) [Kong *et al.*, 2013]: It extracts pairwise social features from partially aligned social networks and then solves the anchor link prediction problem as a classification problem. We use the same settings to extract cross-network social features.
- Collective Random Walk (CRW) [Zhang and Yu, 2015]: Random walk is conducted on networks with anchor links to identify the counterpart of each node in the other network.
- PALE (LIN): PALE model with a linear mapping function in the matching stage.
- PALE (MLP): PALE model with the MLP being employed as the mapping function, where the dimension of the hidden layer is $2 * d$, the learning rate and the regularizing coefficient are chosen based on a 5-fold cross-validation.

As the anchor link prediction algorithms output a list of candidate nodes for each node in the source network, F1-measure and MAP@30 are adopted as metrics for performance comparison.

3.2 Experiment on Facebook Dataset

Dataset

The first dataset[1] was crawled from Facebook and published in [Viswanath *et al.*, 2009]. In the experiment, we filter out those nodes whose degree is less than 5, resulting in $40,710$ users and $766,519$ edges. We then adopt the following process to sample two sub-networks, where nodes are all inherited from the original network. For each edge, we generate a random value p with the uniform distribution in $[0, 1]$. If $p \leq 1-2\alpha_s+\alpha_s\alpha_c$, the edge is discarded; If $1-2\alpha_s+\alpha_s\alpha_c < p \leq 1-\alpha_s$, it is preserved only in the first sub-network; If $1-\alpha_s < p \leq 1-\alpha_s\alpha_c$, it is kept only in the other sub-network; Otherwise, it is preserved in both sub-networks. With such a sample strategy, both sub-networks keep the same sample ratio α_s of edges from the original network in average, reflecting the *sparsity* level of networks. Besides, an expected fraction α_c of edges are shared among the two sub-networks, reflecting the *overlap* level.

In the experiment, one of the two sub-networks is selected as source network G^s, and the other as target network G^t. Since only edges could be different in sub-networks, every corresponding pair of nodes in G^s and G^t are anchor linked as the ground truth, and a fraction α_t of them are chosen as supervised anchor links T. Through the experiment, we demonstrate the performance of the PALE model on different settings of sparsity level α_s and overlap level α_c, compared with all baseline models.

Results and Comparison

To comprehensively evaluate PALE, we compare it with baseline methods in different settings. First, $\alpha_t = 3\%$ anchor links are sampled for training and then predict the rest ones as test data. Experimental results are presented in Table 1. In Table 1(a) different sparsity levels $\alpha_s = [0.5, 0.6, \cdots, 0.9]$ with the same overlap level $\alpha_c = 0.9$ are tested. Meanwhile, with fixed sparsity level $\alpha_s = 0.6$, different overlap levels $\alpha_c = [0.5, 0.6, \cdots, 0.9]$ are examined in Table 1(b).

From Table 1, we can notice that the method using only the degrees of nodes to predict anchor links achieves the poorest performance, where both F1 and MAP are less than 0.1. Without considering observed anchor links as supervision, MAD achieves about 0.39 and 0.41 in F1 and MAP, respectively, in the case of $\alpha_c = 90\%$ overlapping edges between the two networks. It becomes worse when the overlap level decreases. For supervised methods, namely, MNA, CRW, and PALE, to evaluate their performance, we conduct 10 runs of the experiment with sampling $\alpha_t = 3\%$ anchor links for training. The average results of F1 and MAP, as well as their standard deviations, are given in Table 1. We can see that the performance of these supervised methods is better than the unsupervised method. Among all supervised methods, PALE (MLP) outperforms the others in terms of both F1 and MAP under all settings. PALE (LIN) achieves comparable performance with CRW, which uses global structural information via random walks. Since MNA considers only node similarity across networks through local structure information, it achieves much worse performance than CRW and PALE. This again demonstrates that structural information is important for improving the accuracy of anchor link prediction, and our network embedding method offers an effective way to capture the structural regularities of networks in a global and specific way.

In addition, we also examine the effect of different sizes of the supervised anchor link set in the PALE model by varying

[1] http://socialnetworks.mpi-sws.org/data-wosn2009.html

Table 1: Performance comparison between different methods for predicting anchor links on the Facebook dataset.

		Sparsity Level α_s				
Metric	Models	50%	60%	70%	80%	90%
F1	Degree	0.0922	0.0932	0.0945	0.0947	0.0954
	MAD	0.3899	0.3890	0.3893	0.3904	0.3922
	MNA	0.4262 ± 0.0011	0.4290 ± 0.0016	0.4370 ± 0.0015	0.4365 ± 0.0011	0.4390 ± 0.0012
	CRW	0.8693 ± 0.0006	0.8724 ± 0.0012	0.8734 ± 0.0013	0.8786 ± 0.0009	0.8820 ± 0.0022
	PALE (LIN)	0.8652 ± 0.0011	0.8693 ± 0.0004	0.8713 ± 0.0007	0.8752 ± 0.0011	0.8802 ± 0.0012
	PALE (MLP)	$\mathbf{0.8936 \pm 0.0005}$*	$\mathbf{0.8943 \pm 0.0012}$*	$\mathbf{0.8966 \pm 0.0011}$*	$\mathbf{0.9009 \pm 0.0012}$*	$\mathbf{0.9012 \pm 0.0011}$*
MAP	Degree	0.0928	0.0979	0.0983	0.0987	0.0991
	MAD	0.3931	0.4032	0.4097	0.4112	0.4128
	MNA	0.4571 ± 0.0002	0.4573 ± 0.0015	0.4583 ± 0.0013	0.4591 ± 0.008	0.4594 ± 0.0007
	CRW	0.8834 ± 0.0005	0.8835 ± 0.0008	0.8898 ± 0.0004	0.8912 ± 0.0011	0.8915 ± 0.0004
	PALE (LIN)	0.8875 ± 0.0012	0.8881 ± 0.0012	0.8829 ± 0.0005	0.8856 ± 0.0010	0.8881 ± 0.0014
	PALE (MLP)	$\mathbf{0.9100 \pm 0.0008}$*	$\mathbf{0.9207 \pm 0.0009}$*	$\mathbf{0.9224 \pm 0.0011}$*	$\mathbf{0.9228 \pm 0.0005}$*	$\mathbf{0.9237 \pm 0.0008}$*

(a) Experimental results under different sparsity levels, $\alpha_c = 90\%$. Significantly outperforms CRW at the: * 0.01 level, paired t-test.

		Overlap Level α_c				
Metric	Models	50%	60%	70%	80%	90%
F1	Degree	0.0089	0.0128	0.0334	0.0664	0.0932
	MAD	0.1020	0.1523	0.2021	0.3337	0.3890
	MNA	0.1340 ± 0.0012	0.1888 ± 0.0010	0.2170 ± 0.0013	0.3521 ± 0.0015	0.4290 ± 0.0016
	CRW	0.2940 ± 0.0010	0.3823 ± 0.0012	0.5510 ± 0.0012	0.7350 ± 0.0008	0.8724 ± 0.0012
	PALE (LIN)	0.3030 ± 0.0005	0.4079 ± 0.0004	0.5463 ± 0.0007	0.7330 ± 0.0009	0.8693 ± 0.0004
	PALE (MLP)	$\mathbf{0.3789 \pm 0.0005}$*	$\mathbf{0.4518 \pm 0.0009}$*	$\mathbf{0.5914 \pm 0.0010}$*	$\mathbf{0.7714 \pm 0.0011}$*	$\mathbf{0.8943 \pm 0.0012}$*
MAP	Degree	0.0102	0.0133	0.0358	0.0704	0.0979
	MAD	0.1321	0.1633	0.2312	0.3449	0.4032
	MNA	0.1450 ± 0.0012	0.2498 ± 0.0010	0.2923 ± 0.0011	0.3978 ± 0.0012	0.4573 ± 0.0015
	CRW	0.3245 ± 0.0007	0.4133 ± 0.0009	0.5998 ± 0.0014	0.7732 ± 0.0010	0.8835 ± 0.0008
	PALE (LIN)	0.3399 ± 0.0005	0.4432 ± 0.0004	0.5659 ± 0.0009	0.7756 ± 0.0011	0.8881 ± 0.0012
	PALE (MLP)	$\mathbf{0.4197 \pm 0.0010}$*	$\mathbf{0.4845 \pm 0.0007}$*	$\mathbf{0.6224 \pm 0.0012}$*	$\mathbf{0.8118 \pm 0.0010}$*	$\mathbf{0.9207 \pm 0.0009}$*

(b) Experimental results under different overlap levels, $\alpha_s = 60\%$. Significantly outperforms CRW at the: * 0.01 level, paired t-test.

α_t from 0.5% to 5%. Without loss of generality, we check the performance of PALE (MLP) only in terms of MAP. In Figure 3(a), the overlap level is fixed as $\alpha_c = 0.9$, while the sparsity level α_s varies from 0.5 to 0.9. In Figure 3(b), the sparsity level is fixed as $\alpha_s = 0.6$, while the overlap level α_c takes different values in $[0.5, 0.6, ..., 0.9]$. From Figure 3, we can observe that as the size of the training set increases from 0.5% to 2%, the performance of the PALE model is raised quickly under various settings of sparsity level and overlap level. The increase speed slows down when α_t exceeds 2%. That is to say, PALE can lead to good performance with relatively less supervised information. Another interesting phenomenon is that, anchor link prediction between sparser networks can achieve similar prediction results, especially with more supervised anchor links, as Figure 3(a) shows. In Figure 3(b), two networks with less overlap edges, e.g. $\alpha_c = 50\%$, cannot predict as correctly as those with more overlap edges, even when more anchor links are used for training. Thus, it suggests that as compared to sparsity level, overlap level plays a more important role in the performance of anchor link prediction.

3.3 Experiment on Co-author Networks

Dataset

The second dataset used in this paper is a co-author network formed by conference papers from the fields of Artificial Intelligence (AI) and Data Mining (DM), thus denoted as AI-DM, which were extracted from the Microsoft Academic Graph (MAG) [Sinha *et al.*, 2015][2]. MAG is a heterogeneous graph containing bibliographic information of publications, citation relationships between publications, and information of authors and institutions. We choose 10 representative conferences on AI and DM[3], respectively. Two co-author networks are then built on the two groups of papers, and the authors with less than 3 co-author relationships are filtered out. The statistics of the dataset is listed in Table 2. There are $1,154$ common authors between the two networks, forming the ground truth of anchor links.

Table 2: Statistics of the AI-DM dataset.

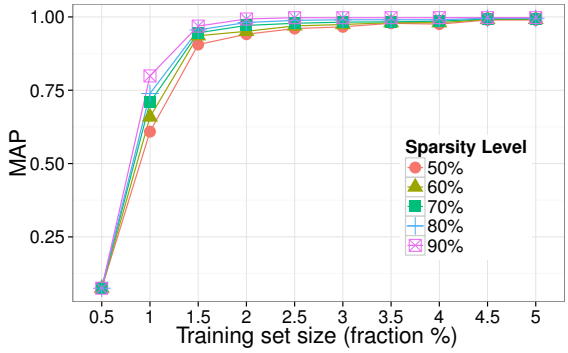
Statistics	AI	DM
$\|V\|$	12,732	15,253
$\|E\|$	45,140	65,993

Results and Comparison

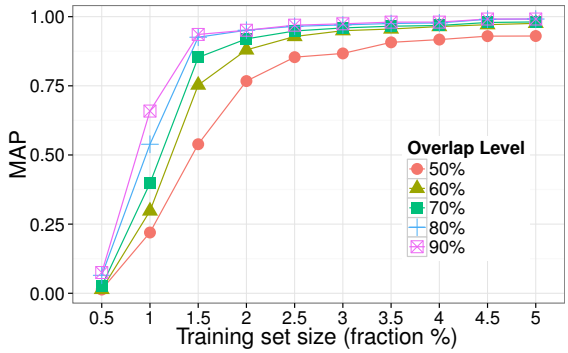
We consider training sets with α_t ranging from 5% to 50%, and the rest of anchor links are used for testing. For each sample ratio α_t, we also carry out 10 runs of the experiment and report the average results as shown in Figure 4. It is shown that PALE exhibits the best performance on predicting anchor links between the two co-author networks on AI and DM. PALE (LIN) performs better than CRW in the cases where the training set is less than 30% of existing anchor links. For example, when 5% of existing anchor links are taken as the training set, PALE (LIN) outperforms CRW by 37% (from

[2]http://research.microsoft.com/en-us/projects/mag/

[3]The conferences selected from the AI field are IJCAI, AAAI, CVPR, ICCV, ICML, NIPS, UAI, ACL, EMNLP, and ECAI, while the conferences selected from the DM field include KDD, SIGMOD, SIGIR, ICDM, ICDE, VLDB, WWW, SDM, CIKM, and WSDM.

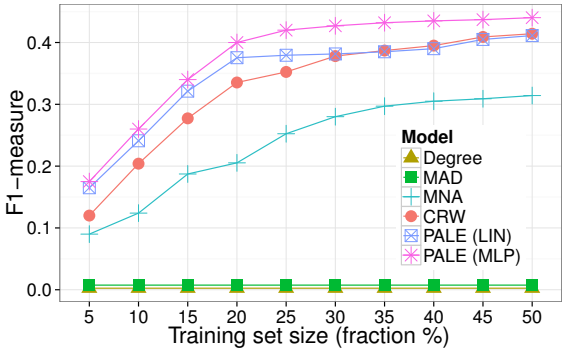


(a) Different sparsity level with overlap level = 0.9

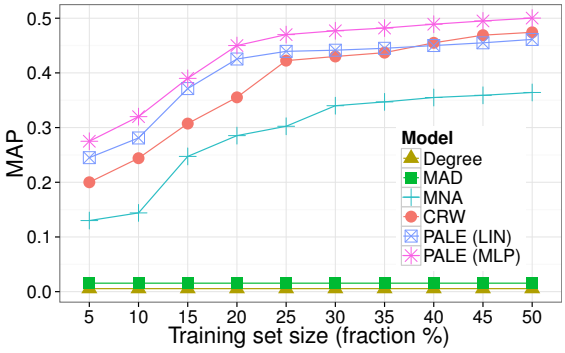


(b) Different overlap level with sparsity level = 0.6

Figure 3: Performance results with different networks sampled from the Facebook dataset.



(a) F1-Measure



(b) MAP

Figure 4: Performance comparison between different methods for predicting anchor links on the AI-DM dataset.

0.1200 to 0.1651). However, when the training fraction α_t is greater than 30%, CRW becomes comparable to or even slightly superior to PALE (LIN), while PALE (MLP) still keeps superior. This suggests that a non-linear function can better capture the mapping relationship across latent spaces. It also justifies the validity of the PALE model.

3.4 Discussion on Network Embedding

In the PALE model, network embedding plays an important role. There are several well-known methods for network embedding, including Spectral Clustering (SC) [Tang and Liu, 2011], Graph Factorization (GF) [Ahmed *et al.*, 2013], Deepwalk [Perozzi *et al.*, 2014]. In this subsection we investigate the performance of the network embedding method proposed in this paper by comparing it with those existing ones. For this purpose, we replace the network embedding method proposed in this paper with those existing ones in the PALE model and then compare the performance of different PALE models, where MLP is taken as the mapping function. Moreover, we want to see how much the contribution of the cross-network extension step to PALE. For this purpose, we compare the performance of PALE with and without cross-network extension, denoted as PALE(-). The experiment is conducted on the AI-DM dataset, and the training set is sampled with $\alpha_t = 30\%$. In Table 3, we present the average values and standard variances of F1 and MAP@30 for 10 runs. It can be seen that our network embedding method outperforms all existing ones.

Table 3: Comparing different network embedding methods.

Methods	F1	MAP
SC	0.4129 ± 0.0011	0.4698 ± 0.0007
GF	0.3455 ± 0.0008	0.3804 ± 0.0005
DeepWalk	0.4010 ± 0.0009	0.4571 ± 0.0008
PALE(-)	0.4093 ± 0.0007	0.4594 ± 0.0007
PALE	$\mathbf{0.4271 \pm 0.0009}$	$\mathbf{0.4770 \pm 0.0011}$

4 Conclusions

In this paper, we proposed an embedding and matching based model for anchor link prediction, called PALE. Different from existing methods that either cope with anchor link prediction as an unsupervised network alignment problem or directly work on structural features defined on networks, the proposed model employs network embedding to keep major structural regularities of networks with awareness of supervised anchor links and then learns a stable cross-network mapping for anchor link prediction. The effectiveness of the proposed model was evaluated on two datasets. As future works, we will devote to designing a unified optimization method for anchor link prediction, balancing the costs at the network embedding and latent space matching stages.

Acknowledgments

This work was funded by the National High-tech R&D Program of China (863 Program) under grant number 2014AA015103, the National Basic Research Program of China (973 Program) under grant numbers 2012CB316303 and 2014CB340401, and the National Natural Science Foundation of China under grant numbers 61425016, 61472400, 61572473, 61572467. Huawei Shen is also funded by Youth Innovation Promotion Association CAS.

References

[Ahmad *et al.*, 2010] Muhammad Aurangzeb Ahmad, Zoheb Borbora, Jaideep Srivastava, and Noshir Contractor. Link prediction across multiple social networks. In *ICDMW '10*, 2010.

[Ahmed *et al.*, 2013] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. In *WWW '13*, 2013.

[Backstrom *et al.*, 2007] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW '07*, 2007.

[Bayati *et al.*, 2009] Mohsen Bayati, Margot Gerritsen, David F Gleich, Amin Saberi, and Ying Wang. Algorithms for large, sparse network alignment problems. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 705–710. IEEE, 2009.

[Cui *et al.*, 2013] Yi Cui, Jian Pei, Guanting Tang, Wo-Shun Luk, Daxin Jiang, and Ming Hua. Finding email correspondents in online social networks. *World Wide Web*, 2013.

[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 *ICDM '12*, 2012.

[Man *et al.*, 2015] Tong Man, Huawei Shen, Junming Huang and Xueqi Cheng. Context-adaptive matrix factorization for multi-context recommendation. In *CIKM '15*, 2015.

[Iofciu *et al.*, 2011] Tereza Iofciu, Peter Fankhauser, Fabian Abel, and Kerstin Bischoff. Identifying users across social tagging systems. In *ICWSM '11*, 2011.

[Klau, 2009] Gunnar W Klau. A new graph-based method for pairwise global network alignment. *BMC bioinformatics*, 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*, 2012.

[Kong *et al.*, 2013] Xiangnan Kong, Jiawei Zhang, and Philip S Yu. Inferring anchor links across multiple heterogeneous social networks. In *CIKM '13*, 2013.

[Li and Lin, 2014] Chung-Yi Li and Shou-De Lin. Matching users and items across domains to improve the recommendation quality. In *SIGKDD '14*, 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 *WSDM '13*, 2013.

[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 *SIGMOD '14*, 2014.

[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 *ASONAM '12*, 2012.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS '13*, 2013.

[Novak *et al.*, 2004] Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Anti-aliasing on the web. In *WWW '04*, 2004.

[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 *ISCID '13*, 2013.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD '14*, 2014.

[Ruck *et al.*, 1990] Dennis W Ruck, Steven K Rogers, Matthew Kabrisky, Mark E Oxley, and Bruce W Suter. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1990.

[Singh *et al.*, 2007] Rohit Singh, Jinbo Xu, and Bonnie Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Research in computational molecular biology*, 2007.

[Sinha *et al.*, 2015] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. In *WWW '15*, 2015.

[Sun *et al.*, 2012] Yizhou Sun, Jiawei Han, Xifeng Yan, and Philip S Yu. Mining knowledge from interconnected data: a heterogeneous information network analysis approach. *Proceedings of the VLDB Endowment*, 2012.

[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 *AAAI '14*, 2014.

[Tang and Liu, 2011] Lei Tang and Huan Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery '11*, 2011.

[Viswanath *et al.*, 2009] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in Facebook. In *WOSN '09*, 2009.

[Zafarani and Liu, 2009] Reza Zafarani and Huan Liu. Connecting corresponding identities across communities. In *ICWSM '09*, 2009.

[Zafarani and Liu, 2013] Reza Zafarani and Huan Liu. Connecting users across social media sites: a behavioral-modeling approach. In *SIGKDD '13*, 2013.

[Zhang and Yu, 2015] Jiawei Zhang and Philip S. Yu. Integrated anchor and social link predictions across partially aligned social networks. In *IJCAI '15*, 2015.

[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 *SIGKDD '15*, 2015.