entropy

MDPI

Article

Anchor Link Prediction across Attributed Networks via Network Embedding

Shaokai Wang [1,2], Xutao Li [3], Yunming Ye [3,*], Shanshan Feng [4], Raymond Y. K. Lau [5], Xiaohui Huang [6] and Xiaolin Du [7]

[1] Guanghua School of Management, Peking University, Beijing 100871, China; wangsk@pku.edu.cn
[2] Harvest Fund Management Co., Ltd., Beijing 100005, China
[3] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China; lixutao@hit.edu.cn
[4] Tencent, Shenzhen 518057, China; sfeng003@e.ntu.edu.sg
[5] Department of Information Systems, City University of Hong Kong, Kowloon Tong, Hong Kong, China; raylau@cityu.edu.hk
[6] School of Information Engineering Department, East China Jiaotong University, Nanchang 330013, China; hxh016@gmail.com
[7] College of Computer Science, Beijing University of Technology, Beijing 100124, China; du_xiaolin@bjut.edu.cn
* Correspondence: yeyunming@hit.edu.cn

Received: 24 January 2019; Accepted: 2 March 2019; Published: 6 March 2019

Abstract: Presently, many users are involved in multiple social networks. Identifying the same user in different networks, also known as anchor link prediction, becomes an important problem, which can serve numerous applications, e.g., cross-network recommendation, user profiling, etc. Previous studies mainly use hand-crafted structure features, which, if not carefully designed, may fail to reflect the intrinsic structure regularities. Moreover, most of the methods neglect the attribute information of social networks. In this paper, we propose a novel semi-supervised network-embedding model to address the problem. In the model, each node of the multiple networks is represented by a vector for anchor link prediction, which is learnt with awareness of observed anchor links as semi-supervised information, and topology structure and attributes as input. Experimental results on the real-world data sets demonstrate the superiority of the proposed model compared to state-of-the-art techniques.

Keywords: anchor link prediction; network embedding; attributed network

1. Introduction

In recent years, with the popularity of various social network platforms, a user is usually involved in multiple social networks simultaneously [1]. Due to the function diversity, users in different platforms may express their opinions on various topics, share distinct types of content or follow different users. For example, a user may use Facebook to follow and share entertainment news, and use Quora to gain and share knowledge. The social network platforms profile users from different points of view. If we can identify the same user in different social networks, her profile can be better characterized for a more accurate classification or recommendation. The problem of identification of the same users across multiple networks is known as anchor link prediction, where the associations are termed as anchor links [2].

Despite great application values, solving the problem is challenging, because of the complex network structures, the rich attribute information, and few observed anchor links. Early studies mainly solve the problem by exploiting user profiles (e.g., user name, location, gender) [3,4], demographical features [5] or user generated contents, such as, tweets, posts and reviews [6]. Recently, network structures are also leveraged to address the problem. Particularly, methods of this type rely on hand-crafted network features. For example, match degree can be computed using the number of shared identified friends [7], in/out neighbors and in/out degree [8], and Dice coefficient [9]. However, the hand-crafted features only partially reflect the intrinsic structure regularities of networks, thereby producing less satisfactory performance for anchor link prediction.

Recently, network-embedding techniques have been used to learn latent network features, which can better preserve the structural regularity of a network [10,11]. To match users in different social platforms, Liu et al. [12] proposed IONE. The algorithm learns the embedding of a user by predicting her network contexts, i.e., the follower-ship/followee-ship, where observed anchor links are used to transfer contexts between networks. Man et al. [13] proposed a supervised model PALE, which maps each user into a low dimension space for the identification of anchor links. However, the two models use only the network topology without considering node attributes, which are not applicable to attributed networks. Moreover, in practice the observed anchor links are very few and thus a semi-supervised solution is quite advisable and desirable.

In this paper, we propose a novel semi-supervised model (APAN) to tackle the Anchor lLink Prediction across Attributed Networks. The key idea of APAN is learning the embedding of each network to represent its users into a low-dimensional space. On the one hand, the low-dimensional representation of each user is used to predict her contexts in the network, i.e., the random-walk sequences generated via network topology and the neighboring nodes that share same attribute information. On the other hand, it is leveraged to predict the anchor links (semi-supervised information) between networks. By doing so, the nodes (users) that have similar structure contexts and attributed information will be close in the embedding space. Also, the anchor link predictor is simultaneously trained as the embeddings are learnt. The real-world data sets are used to evaluate the performance of the proposed APAN model. Experimental results show that APAN outperforms state-of-the-art competitors.

2. Related Work

Recently, network embedding has aroused a lot of research interest. Network embedding aims to learn low-dimensional representations of network nodes, while effectively preserving network topology structure, node content, and other side information. Inspired by the idea of word representation learning [14], Perozzi el al. [10] developed DeepWalk to learn the representations of nodes in a network, which can preserve the neighbor structures of nodes. Node2vec [15] further exploits a biased random-walk strategy to capture more flexible contextual structure. Network structures include first-order structure and higher-order structure. LINE [11] is proposed to preserve the first-order and second-order proximities. The first-order proximity is the observed pairwise proximity between two nodes. The second-order proximity is determined by the similarity of the neighbors of two nodes. Besides network structures, node content is another important information source for network embedding. With content information incorporated, the learnt node representations are expected to be more informative. Yang et al. [16] propose TADW that takes the rich information (e.g., text) associated with nodes into account when they learn the low-dimensional representations of nodes. Pan et al. [17] propose TriDNR which is a coupled deep model that incorporates network structure, node attributes, and node labels into network embedding. LANE [18] is also proposed to incorporate the label information into the attributed network embedding. The task of linking users accounts on multiple social networks, is a challenging task, because social network structures for a specific user can be rather diverse on different social media platforms. Since the traditional network-embedding methods are designed for single network, they cannot handle the anchor link prediction problem. Moreover, the network embeddings are usually learnt in an unsupervised manner, and hence cannot leverage the observed anchor links in anchor link prediction.

Conventional methods for finding correspondence between networks can be mainly divided into two categories. The first category is called network alignment. It works in an unsupervised manner and does not leverage the existing correspondence. Specifically, the type of methods aligns nodes by finding structural similarity between nodes across networks. Network alignment has been widely used in many fields such as bioinformatics [19], computer vision [20], database matching [21], etc. However, ignoring the observed correspondence is obviously a waste of knowledge. The second category belongs to supervised methods, which learns a predictor relying on the observed anchor links [22]. Most of studies train the predictor directly using the hand-crafted network features, such as common neighbors [7], degree [8], clustering coefficient [9], etc. However, the hand-crafted features may not capture all the intrinsic structural regularities of the networks, thereby producing less satisfactory performance.

With the advancement of deep learning, network-embedding techniques are developed to identify the same users in different platforms. For example, Liu et al. [12] proposed IONE algorithm, which embeds users into a low-dimensional space for anchor link prediction. Man et al. [13] proposed an embedding and matching-based model PALE. However, different from our approach, the network embedding in PALE is purely unsupervised and does not leverage observed anchor links when encoding the network structure into embeddings. Moreover, the two approaches cannot make use of network attributes. Recently, Zhang et al. [23] proposed an attributed network alignment algorithm, called FINAL. The method leverages the node attribute information to guide the topology-based alignment. In FINAL, a nice alignment consistency principle is designed and developed, i.e., the alignments between two pairs of nodes across the networks should be "similar/consistent" with each other. However, this algorithm works in an unsupervised manner and cannot leverage the observed anchor links.

3. Methods

3.1. Problem Formulation

Assume we are given an attributed network $G = (X, E, A)$, where $X = \{x_1, \ldots, x_N\}$ is a set of nodes, E is the adjacent matrix, E_{ij} is the weight of the edge between nodes x_i and x_j. If there is a connection between x_i and x_j, $E_{ij} = 1$, otherwise $E_{ij} = 0$. $A = \{a_1, \ldots, a_N\}$ denotes the attributes of N nodes. The scenario considered here is that one user has two accounts registered on two different social networks, and the two accounts are connected through an anchor link. Without loss of generality, we use one network as source network and the other as target network, denoted with G^s and G^t respectively. As shown in the Figure 1, some anchor links are already known between G^s and G^t. For each node that has no anchor links in the source network G^s, the purpose of this paper is to find its corresponding node in the target network G^t. This can be formalized as the following anchor link prediction problem:

Definition 1. *(**Anchor Link Prediction**) Given two attributed networks $G^s = (X^s, E^s, A^s)$ and $G^t = (X^t, E^t, A^t)$, and the existing anchor links $T = \{(x^s, x^t) | x^s \in G^s, x^t \in G^t\}$. The anchor link prediction problem is to predict potential anchor links across G^s and G^t.*

As aforementioned in the introduction, our approach consists of two important components: one is the attributed network embedding and the other is the semi-supervised anchor link predictor. Next, we will introduce our APAN approach by elaborating them, respectively.

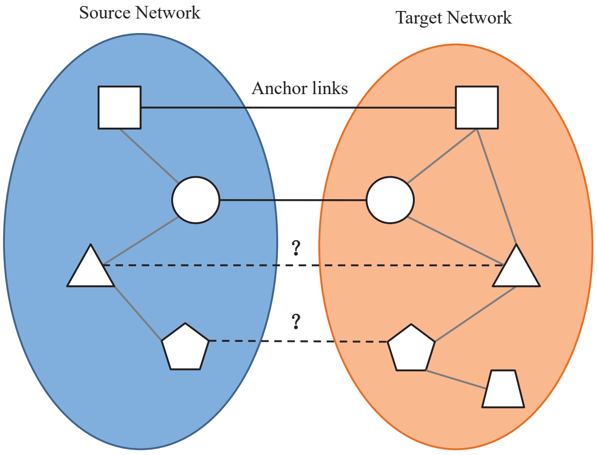


Figure 1. The anchor link prediction problem.

3.2. Learning Attributed Network Embedding

Skip-gram is a popular framework of embedding representation learning [14], which was first developed to capture the word semantic correlations. Given a word and its context $\{(x_i, x_c)\}$, x_i denotes the current word, and the context x_c is a neighbor word around x_i within a fixed window size. Skip-gram uses the embedding vector $\mathbf{e}_i$ of word x_i as input feature, and then predicts its context x_c by minimizing the following log loss function:

$$-\sum_{(x_i,x_c)} \log P(x_c|x_i) = -\sum_{(x_i,x_c)} \left(\mathbf{w}_c^T \mathbf{e}_i - \log \sum_{c' \in \mathbb{C}} \exp(\mathbf{w}_{c'}^T \mathbf{e}_i)\right) \tag{1}$$

where $\mathbb{C}$ denotes the entire context space, which includes all the vocabularies of the corpus; $\{\mathbf{w}_c\}_{c \in \mathbb{C}}$ are the model parameters.

Inspired by Skip-gram, Perozzi et al. [10] developed DeepWalk to model the node correlation from the topology point of view. In DeepWalk, the embedding vector of a node is used to predict its network context, i.e., the node sequences generated by random walk regarding the node. Specifically, in each training node pair (x_i, x_c), x_i is the current node, and the context x_c is each of the neighboring nodes within a fixed window size regarding x_i in the random-walk sequences. Here the context space $\mathbb{C}$ includes all the nodes in the network.

Next, we introduce how to extend the idea of DeepWalk for attributed network embedding. Let first assume that the embeddings for the source network G^s and the target network G^t are learnt independently here (In next subsection, we will discuss how to build connections between them). Suppose each node in the two networks is embedded into an e-dimensional vector.

To design the attributed network-embedding algorithm, we first need to understand the optimization strategy for Equation (1). A direct optimization to the objective is costly, because the second term must be normalized over the entire context space $\mathbb{C}$, which is huge. In [14], a negative sampling strategy is used to tackle the problem. Specifically, the method re-casts the normalization-based optimization problem into a sampling-based binary classification problem. Assume (x_i, x_c, γ) is a random sample drawn from a given probability distribution $P(x_i, x_c, \gamma)$. Here x_i and x_c represent the current node and a context, respectively. $\gamma = +1$ represents (x_i, x_c) is a positive pair, where x_c is a node in the context of x_i. $\gamma = -1$ represents (x_i, x_c) is a negative pair that x_c is not in the context of x_i. Given (x_i, x_c, γ), we aim to minimize the cross-entropy loss to the binary class γ:

$$-\mathbb{I}(\gamma = 1) \log \sigma(\mathbf{w}_c^T \mathbf{e}_i) - \mathbb{I}(\gamma = -1) \log \sigma(-\mathbf{w}_c^T \mathbf{e}_i) \tag{2}$$

where σ is the sigmoid function, defined as $\sigma(x) = 1/(1 + e^{-x})$; $\mathbb{I}(\cdot)$ is the indicator function; when the argument is true it outputs 1, otherwise it outputs 0. As the samples follow the distribution $P(x_i, x_c, \gamma)$, the overall loss function can thus be expressed as:

$$- E_{(x_i, x_c, \gamma)} \log \sigma(\gamma \mathbf{w}_c^T \mathbf{e}_i) \tag{3}$$

where E indicates the expectation operator.

In our attributed network-embedding scenario, the key issue now becomes generating samples with the distribution $P(x_i, x_c \gamma)$. We give the concrete implementation in Algorithm 1. Two types of contexts are sampled in the algorithm. The first type is based on the network structure and the second type based on the node attributes A. By doing so, the learnt embeddings not only reflect the structure context, but also the attribute context.

Algorithm 1 Sampling context algorithm

```
Input: Network $G$, node attributes $A$, parameters $r_1$, $r_2$, $q$, $e$ and $d$;
Output: $(x_i, x_c, \gamma)$;
1: if $random_1 < r_1$ then
2:     $\gamma \leftarrow +1$;
3: else
4:     $\gamma \leftarrow -1$;
5: end if
6: if $random_2 < r_2$ then
7:     Uniformly sample a random-walk sequence $S$ of length $q$;
8:     if $\gamma = +1$ then
9:         Under the condition $|i - c| < d$, sample $(x_i, x_c)$ in $S$;
10:    else
11:        Sample $x_c$ in $\mathbb{C}$;
12:    end if
13: else
14:    if $\gamma = +1$ then
15:        Uniformly sample $(x_i, x_c)$ which satisfies $a_i = a_c$;
16:    else
17:        Uniformly sample $(x_i, x_c)$ which satisfies $a_i \neq a_c$;
18:    end if
19: end if
```

In the algorithm, we use a parameter $r_1 \in (0, 1)$ to control the proportion of positive and negative nodes, and use a parameter $r_2 \in (0, 1)$ to control the ratio of two types of contexts. As shown in lines 1∼5, we first determine whether to sample a positive sample or negative sample, in terms of r_1. Then in line 6, we generate a random number to determine whether to sample from the structure or the attribute context. If the number is smaller than r_2, structure context is chosen. We first produce a random-walk sequence S in line 7. If our previous decision is to sample a positive example, we produce the context x_c such that it is within the window size of d regarding x_i (lines 8∼10), otherwise we randomly choose an example from $\mathbb{C}$ (lines 10∼12). When sampling attribute context (lines 13∼19), positive examples are randomly chosen from the nodes that have the same attribute values, while negative examples are from the ones that have different attribute values.

3.3. Semi-Supervised Anchor Link Prediction

In the subsection, we introduce how to use the observed anchor links for the embedding learning. Given a potential anchor link pair $(x_l^s, x_n^t) \in T$ and the corresponding embedding vectors $\mathbf{e}_l^s$ and $\mathbf{e}_n^t$, the probability that the anchor link exists can be expressed as:

$$P(x_l^s, x_n^t) = \sigma({\mathbf{e}_l^s}^T \cdot \mathbf{e}_n^t) = 1/(1+e^{-{\mathbf{e}_l^s}^T \cdot \mathbf{e}_n^t}) \tag{4}$$

where σ is sigmoid function. To capture more complex associations, we can build a k layers feed-forward neural network as our predictor. The k-th layer h^k of the neural network is a nonlinear function of the previous hidden layer h^{k-1}, defined as

$$h^k(\mathbf{e}) = ReLU(\mathbf{W}^k h^{k-1}(\mathbf{e}) + b^k) \tag{5}$$

where $ReLU(x) = \max(x, 0)$, $\mathbf{W}^k$ and b^k are parameters of k-th layer, and $h^0(\mathbf{e}) = \mathbf{e}$. By using the complex predictor, Equation (4) is rewritten as:

$$\begin{aligned} P(x_l^s, x_n^t) &= \sigma(h^k(\mathbf{e}_l^s)^T \cdot h^k(\mathbf{e}_n^t)) \\ &= 1/(1+e^{-h^k(\mathbf{e}_l^s)^T \cdot h^k(\mathbf{e}_n^t)}) \end{aligned} \tag{6}$$

Combining Equation (6) with Equation (3), we obtain the following objective function for the anchor link prediction problem:

$$\begin{aligned} -\sum_{(x_l^s, x_n^t) \in T} \log P(x_l^s, x_n^t) - \lambda_1 E_{\{(x_i^s, x_c^s, \gamma) | x_i^s, x_c^s \in G^s\}} \log \sigma(\gamma {\mathbf{w}_c^s}^T \mathbf{e}_i^s) \\ - \lambda_2 E_{\{(x_i^t, x_c^t, \gamma) | x_i^t, x_c^t \in G^t\}} \log \sigma(\gamma {\mathbf{w}_c^t}^T \mathbf{e}_i^t) \end{aligned} \tag{7}$$

where λ_1 and λ_2 are two parameters. In Equation (7), the first item is the loss of anchor link prediction, the second item is the loss of context predictions in source network G^s, and the third item is the loss of context predictions in target network G^t. The network structure of APAN algorithm is shown in Figure 2.

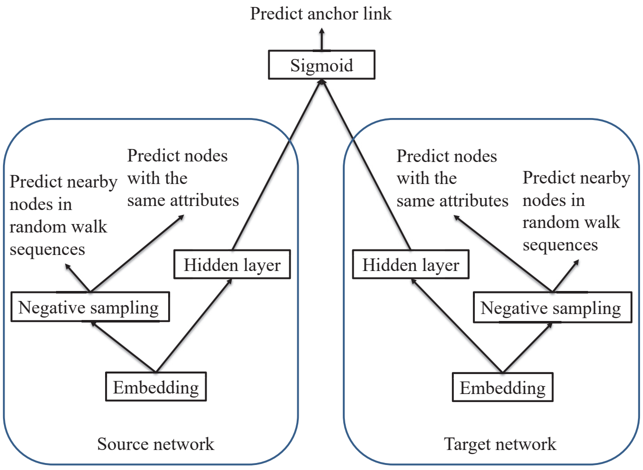

Predict anchor link
Sigmoid
Predict nearby nodes in random walk sequences
Predict nodes with the same attributes
Hidden layer
Negative sampling
Embedding
Source network
Predict nodes with the same attributes
Predict nearby nodes in random walk sequences
Hidden layer
Negative sampling
Embedding
Target network

Figure 2. APAN network architecture.

By optimizing Equation (7), we ultimately obtain the embedding vectors $\mathbf{e}$ of all nodes in the source network G^s and the target network G^t. When predicting anchor links, given a node x_l^s in the source network G^s, we can calculate the probabilities that x_l^s has anchor links with all the nodes in the target network G^t, by using Equation (6). Sorting them in terms of the probabilities offers us a list of potential anchor links.

When training the proposed APAN, a stochastic gradient descent in the mini-batch mode is adopted [24]. In each iteration, a set of node pairs in the anchor links set T is first sampled and a gradient calculation is performed to optimize the loss of anchor link prediction. Subsequently, we sample a set of context (x_i^s, x_c^s, γ) in the source network G^s and perform a gradient calculation to optimize the loss function of predicting context in G^s. Similarly, we sample a set of context (x_i^t, x_c^t, γ) in the target network G^t and perform a gradient calculation to optimize the loss of predicting context in G^t. The model training procedure is implemented as Algorithm 2.

Algorithm 2 Model training

```
Require: Attributed networks $G^s$ and $G^t$, parameters $\lambda_1$ and $\lambda_2$, batch sizes $K_1$, $K_2$ and $K_3$;
for $it = 1 \rightarrow Max\_It$ do
    Sample a group of node pairs of size $K_1$ in the anchor links set $T$;
    Let $\mathbb{R}_1 = -\frac{1}{K_1}\sum_{(x_l^s, x_n^t)} \log P(x_l^s, x_n^t)$, perform a gradient calculation on $\mathbb{R}_1$;
    Sample a group of contexts $(x_i^s, x_c^s, \gamma)$ of size $K_2$ in the source network $G^s$;
    Let $\mathbb{R}_2 = -\frac{\lambda_1}{K_2}\sum_{(x_i^s, x_c^s, \gamma)} \log \sigma(\gamma \mathbf{w}_c^{s\,T} \mathbf{e}_i^s)$, perform a gradient calculation on $\mathbb{R}_2$;
    Sample a group of contexts $(x_i^t, x_c^t, \gamma)$ of size $K_3$ in the target network $G^t$;
    Let $\mathbb{R}_3 = -\frac{\lambda_2}{K_3}\sum_{(x_i^t, x_c^t, \gamma)} \log \sigma(\gamma \mathbf{w}_c^{t\,T} \mathbf{e}_i^t)$, perform a gradient calculation on $\mathbb{R}_3$.
end for
```

4. Experiments

In this section, we conduct experiments to compare the proposed APAN algorithm with state-of-the-art techniques.

4.1. Datasets and Baselines

In the experiments, we use three real-world attributed networks, which are Flickr and Lastfm datasets from [25], and Douban dataset from [26]. Following [23], we adopt the following ways to construct our datasets (Table 1).

Table 1. Statistics of the datasets.

Datasets	#Nodes	#Edges
Flickr	4935	15,884
Lastfm	4496	10,628
Douban Online	3906	16,328
Douban Offline	1118	3022

Flickr vs. Lastfm. We extract the subnetworks from Flickr and Lastfm, which contain 4935 nodes and 4496 nodes, respectively. The edges in the two networks are who-follow-whom relationship. We consider the gender of a user as node attribute. For the users whose gender information is missing, we fill in the values of 'unknown'.

Douban Online vs. Douban Offline. The offline network is constructed according to users' co-occurrence in social gatherings. There is an edge in the offline network between two users if they participate in the same offline events more than ten times. The constructed offline network includes 1118 users and we extract a subnetwork with 3906 nodes from the provided online network that contains all these offline users. We treat the location of a user as the node attribute.

We compare APAN algorithm with the following baselines:

- PALE [13]: This algorithm is a network-embedding-based anchor link prediction algorithm. PALE employs network embedding with awareness of observed anchor links as supervised information to capture the structural regularities and further learns a stable cross-network mapping for anchor link prediction.

- ULink [27]: ULink is a projection algorithm designed based on latent user space modelling. They build the latent user space through projection matrix.
- FINAL [23]: FINAL is proposed to solve the attributed network alignment problem. It leverages the node attribute information to guide (topology-based) alignment process.
- APAN-N: This algorithm is a variant of our proposed APAN algorithm. When predicting context using negative sampling, APAN-N only predicts context based on network structure and does not use nodes' attributes.

In the comparison, we implemented the PALE method and use the original implementations of ULink and FINAL methods.

For the anchor link prediction problem, the widely used evaluation metric is to compare the top-*k* ranking list of a potential matching account. The higher the rank of the ground-truth account in the list, the better. In this paper, we evaluate all methods by computing top-*k* precision [27] for each test user as follows:

$$h(x) = \frac{k - (hit(x) - 1)}{k} \tag{8}$$

where $hit(x)$ represents the position of ground-truth account in the returned top-k users. We report the average precision of all the tested users x_i as the result: $\sum_{i=1}^{N} h(x_i)/N$, which is denoted by "Hit-precision".

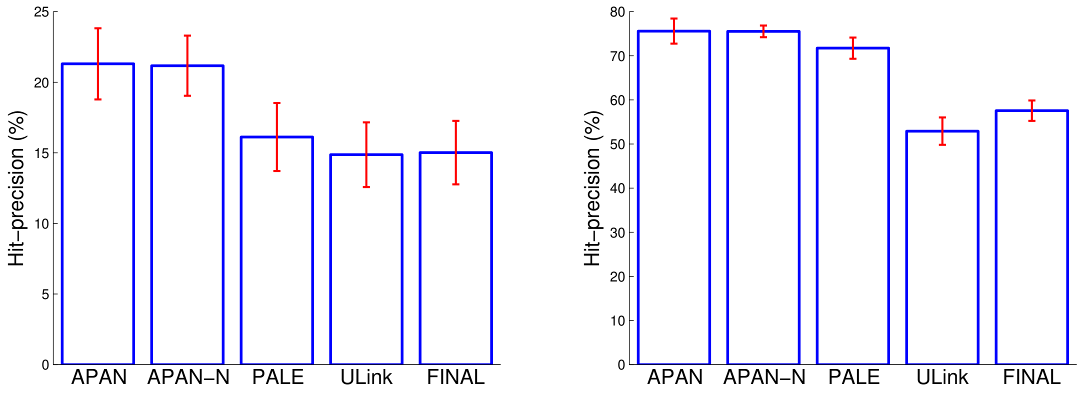
In our experiments, we randomly partition the ground-truth anchor links into five groups and conduct five-fold cross-validation and report the average results. We set the model parameters to $r_1 = 2/3$, $q = 10$, $e = 50$, $d = 3$, $K_1 = 1000$, $K_2 = 2000$ and $K_3 = 2000$. We found that our model is not very sensitive to these parameters. We tune r_2, λ_1 and λ_2 via cross-validation method.

4.2. Experimental Results

Table 2, Figures 3 and 4 show the Hit-precision@*k* results of different compared algorithms with different *k*. From the Figures, we can easily judge the performance trend when varying the number labeled data, whereas the detailed performed can be easily observed from the table. As can be seen from the experimental results, APAN and APAN-N achieve better performance than the baseline methods in most cases. Specifically, APAN outperforms PALE, ULink and FINAL by more than 6%. For instance, the Hit-precision of APAN is 19.42% while the result of PALE is 13.65% when *k* equals to 10 for Flickr-Lastfm networks; the Hit-precision of APAN is 46.53% while the result of PALE is 39.98% when *k* equals to 1 for Douban online-offline networks. Moreover, we observe that APAN which uses node attributes yields better performance than APAN-N that leverages only the network structure. The observation suggests that APAN can effectively exploit network structure and node attributes. This implies that the node attributes and network structure contain useful information to give a comprehensive view about the user. An effective model for network data should thus consider both the node attributes and network structure in the anchor link prediction task. Also, we find that the results of APAN and APAN-N are very close. This is because there are many missing values in the node attributes. In the datasets, the gender and the location are used as attributes of the nodes. For the users whose attributes information are missing, we fill in the values of 'unknown'. The loss of attribute information degenerates the performance of APAN method. We find that network-embedding-based methods APAN, APAN-N, and PALE deliver better results than the other methods. In particular, an accuracy improvement of 17% against other algorithms is observed when $k = 5$ in Douban online-offline networks (PALE with Hit-precision 61.66% versus ULink with 43.93%). The observation demonstrates the effectiveness and merits of network-embedding methods. Compared to the conventional approaches, network-embedding-based methods represent each node into a continuous real-value vector. By doing so, the network structure regularities and attribute information can be summarized into the vector, which is better than the hand-crafted features in conventional methods.

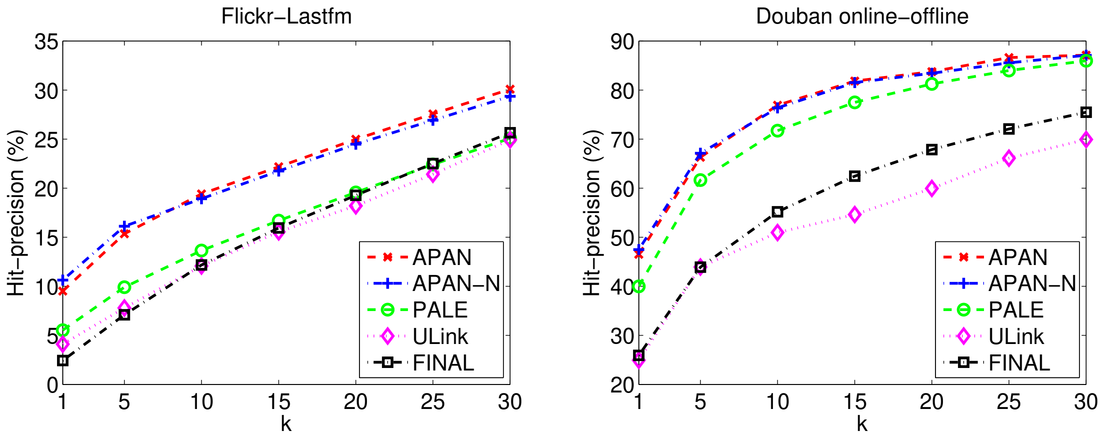
Table 2. Performance of different models on real datasets (%), boldface indicates the best performance.

Dataset	Evaluation Metric	APAN	APAN-N	PALE	ULink	FINAL
Flickr-Lastfm	Hit-p@1	9.51 ± 2.82	**10.63 ± 1.17**	5.53 ± 1.57	4.10 ± 1.22	2.43 ± 0.90
	Hit-p@5	15.35 ± 2.58	**16.12 ± 2.25**	9.91 ± 2.73	7.79 ± 2.06	7.10 ± 0.60
	Hit-p@10	**19.42 ± 2.73**	18.93 ± 2.37	13.65 ± 3.00	12.07 ± 1.92	12.18 ± 1.76
	Hit-p@15	**22.18 ± 2.70**	21.75 ± 2.41	16.70 ± 2.81	15.55 ± 3.24	15.95 ± 2.45
	Hit-p@20	**24.98 ± 2.57**	24.48 ± 2.41	19.57 ± 2.51	18.20 ± 2.91	19.28 ± 3.04
	Hit-p@25	**27.57 ± 2.25**	26.92 ± 2.23	22.42 ± 2.20	21.42 ± 2.88	22.51 ± 3.28
	Hit-p@30	**30.09 ± 1.94**	29.36 ± 2.04	25.08 ± 2.07	24.93 ± 1.81	25.65 ± 3.63
Douban online-offline	Hit-p@1	46.53 ± 2.80	**47.49 ± 2.98**	39.98 ± 3.30	24.95 ± 3.05	25.93 ± 2.17
	Hit-p@5	66.31 ± 3.23	**67.12 ± 1.29**	61.66 ± 2.04	43.93 ± 4.03	43.90 ± 1.20
	Hit-p@10	**76.91 ± 3.09**	76.38 ± 0.95	71.72 ± 2.06	50.95 ± 2.74	55.19 ± 2.27
	Hit-p@15	**81.84 ± 2.79**	81.54 ± 0.98	77.49 ± 2.24	54.62 ± 1.85	62.44 ± 2.81
	Hit-p@20	**83.76 ± 2.39**	83.43 ± 1.00	81.25 ± 2.24	59.94 ± 2.36	67.86 ± 2.77
	Hit-p@25	**86.62 ± 2.12**	85.55 ± 1.12	83.99 ± 2.17	66.11 ± 3.22	72.08 ± 2.64
	Hit-p@30	**87.12 ± 1.99**	87.09 ± 1.03	85.98 ± 2.14	69.92 ± 4.39	75.49 ± 2.34



(**a**) (**b**)

Figure 3. The average Hit-precision@*k* performance on real datasets. (**a**) Flickr-Lastfm networks; (**b**) Douban online-offline networks.



(**a**) (**b**)

Figure 4. Hit-precision@*k* performance on real datasets. (**a**) Flickr-Lastfm networks; (**b**) Douban online-offline networks.

Next, we compare APAN-N and PALE algorithms, which are all based on network-embedding learning and use only network structure. We can see APAN-N performs better than PALE in most cases. APAN-N achieves an accuracy improvement of 6% against PALE when k equals to 5 for Douban online-offline networks. There are two main reasons. On the one hand, the proposed APAN-N works in a semi-supervised manner. During the training phase, three objectives, namely, the anchor link prediction, the context prediction in the source network and the one in the target network, are iterated. Hence, the produced node embeddings incorporate both the supervised and unsupervised information. However, PALE breaks up the network-embedding learning and exist anchor link prediction into two independent phases. As a result, the node embedding vectors produced by PALE are only related to the network structure. Hence, the embeddings produced by our APAN-N are more helpful for the anchor link prediction task. On the other hand, PALE uses the first-order proximity structure in the network-embedding learning process [11]. The method only models the local adjacency of each node, but ignores the global connection property in the network. Therefore, PALE is not sufficient to preserve the intrinsic structure regularities of networks. Instead, our APAN-N uses the truncated random-walk sequences to learn the embeddings, which can capture both the local and global structure properties. This can also be verified by Figure 5, which depicts the performance changes of APAN-N and PALE. A big gap can be found as the number of iterations increases. Due to the reasons, APAN-N works better than PALE.

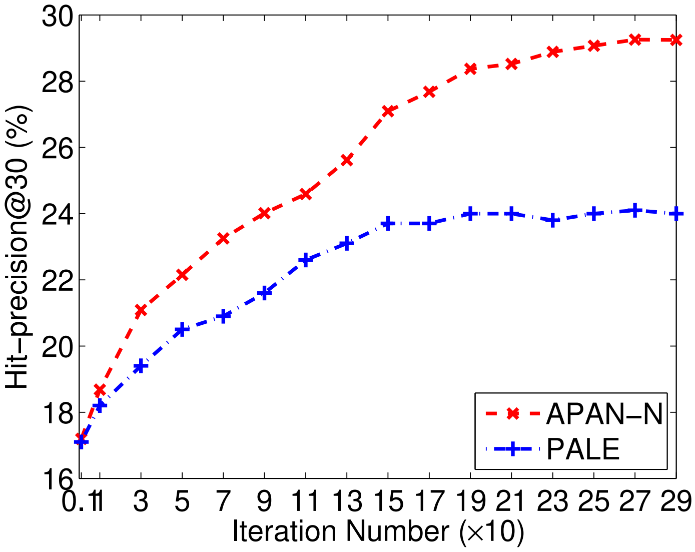


Figure 5. Comparison of APAN-N and PALE in different iterations on Flickr-Lastfm networks.

4.3. Parameter Study

In this subsection, we investigate how different values of the parameter r_2 and the dimension of the embedding vectors affect the performance of APAN.

For our proposed APAN method, we use parameter r_2 to control the ratio of two types of contexts. The larger r_2 is, the more important network structure is. Figure 6 shows the Hit-precision@30 of APAN using different values of parameter r_2 on Flickr-Lastfm networks. From the figure, we observe that the accuracy is very low when r_2 is small, and the accuracy increases when r_2 becomes large. It achieves good performance with the r_2 varying from 0.8 to 0.9. The large value of r_2 indicates the importance of network structure. In the datasets, the node attribute includes the gender and the location. Since there are many missing values in the attributes, and these two kinds of attribute are not strong enough to link users, the structural information is more discriminative than the attribute information.

We investigate the sensitivity of the dimension of the embedding vectors. Figure 7 shows the Hit-precision@30 of APAN with various dimensions on Flickr-Lastfm networks. We observe the performance is poor when the dimensionality is under 30. APAN reaches a relatively stable and promising performance after the dimensionality is higher than 50. This indicates that APAN model is robust with the tuning of dimensions.

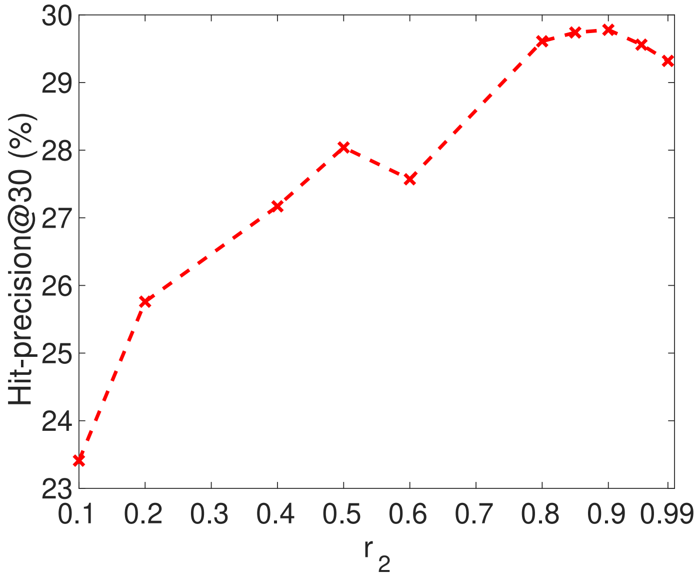


Figure 6. The tuning of r_2 on Flickr-Lastfm networks.

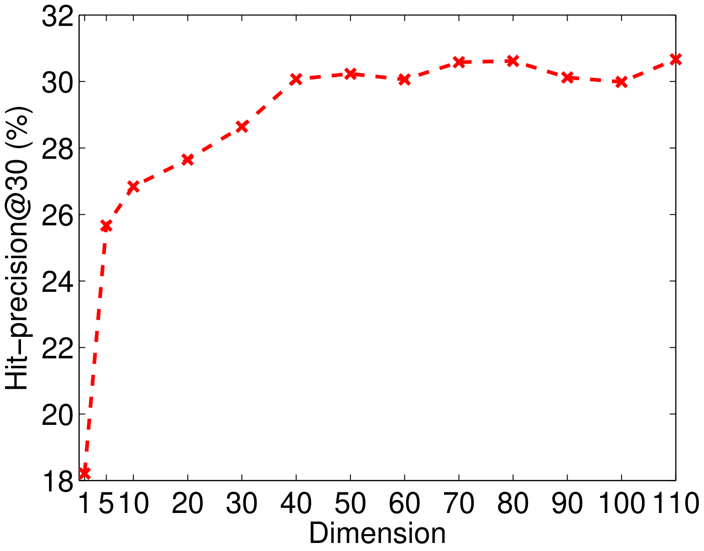


Figure 7. Tuning the dimension of the embedding vectors on Flickr-Lastfm networks.

5. Conclusions

In this paper, we propose a novel semi-supervised network-embedding model (APAN) to tackle the anchor link prediction across attributed networks. APAN represents each node (user) of the multiple networks by a low-dimensional vector, which is learnt with awareness of observed anchor links as semi-supervised information, and topology structure and attributes as input. By doing so, the nodes that have similar structure contexts and attributed information will have similar embedding vectors. Also, the anchor link predictor is simultaneously trained as the embeddings are learnt. The real-world data sets are used to evaluate the performance of the proposed APAN model. Experimental results show that APAN outperforms state-of-the-art competitors.

APAN has two limitations. Firstly, since the node embedding vectors produced by APAN are related to the network structure, the accuracy will be low when the topology structures of the two networks are widely distinct. Secondly, social networks are dynamically changing over time.

The APAN method cannot extract features dynamically. Our next work will solve the above two problems, we may consider integrating more types of information, such as the temporal information, into APAN so that the method can be more robust, and develop a dynamic anchor link prediction algorithm to take advantage of incremental data for improving the performance.

Author Contributions: methodology, S.W., X.L., Y.Y. and S.F.; software, S.W.; investigation, S.W., X.L. and Y.Y.; writing—original draft preparation, S.W.; writing—review and editing, X.L. and S.F.; supervision, Y.Y. and X.H.; funding acquisition, S.W., Y.Y., X.H. R.L. and X.D.

Funding: This research was funded by the NSFC under Grant No. U1836107, 61602132, 61562027 and 61572158, the Natural Science Foundation of Jiangxi Province under Grant No. 20181BAB202024, and Education Department of Jiangxi Province under Grant No. GJJ170413, China Postdoctoral Science Foundation (Grant No. 2018M640368). Lau's work is supported by grants from the RGC of the Hong Kong SAR (Projects: CityU 11502115 and CityU 11525716), the NSFC Basic Research Program (Project: 71671155), and the CityU Shenzhen Research Institute.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Manikonda, L.; Meduri, V.V.; Kambhampati, S. Tweeting the Mind and Instagramming the Heart: Exploring Differentiated Content Sharing on Social Media. In Proceedings of the International Conference on Weblogs and Social Media, Cologne, Germany, 17–20 May 2016; pp. 639–642.
2. Shu, K.; Wang, S.; Tang, J.; Zafarani, R.; Liu, H. User Identity Linkage across Online Social Networks: A Review. *ACM SIGKDD Explor. Newslett.* **2017**, *18*, 5–17. [CrossRef]
3. Liu, J.; Zhang, F.; Song, X.; Song, Y.I.; Lin, C.Y.; Hon, H.W. 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, Rome, Italy, 4–8 February 2013; pp. 495–504.
4. Zhang, J.; Yu, P.S. Multiple Anonymized Social Networks Alignment. In Proceedings of the IEEE International Conference on Data Mining, Atlantic City, NJ, USA, 14–17 November 2015; pp. 599–608.
5. Malhotra, A.; Totti, L.; Meira, W., Jr.; Kumaraguru, P.; Almeida, V. Studying User Footprints in Different Online Social Networks. In Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Beijing, China, 17–20 August 2013; pp. 1065–1070.
6. Novak, J.; Raghavan, P.; Tomkins, A. Anti-aliasing on the web. In Proceedings of the 13th International Conference on World Wide Web, Beijing, China, 18–20 October 2004; pp. 30–39.
7. Zhou, X.; Liang, X.; Zhang, H.; Ma, Y. Cross-Platform Identification of Anonymous Identical Users in Multiple Social Media Networks. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 411–424. [CrossRef]
8. Narayanan, A.; Shmatikov, V. De-anonymizing Social Networks. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 17–20 May 2009; pp. 173–187.
9. Sergey, B.; Anton, K.; Seungtaek, P.; Wonho, R.; Hyungdong, L. Joint link-attribute user identity resolution in online social networks. In Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, Workshop on Social Network Mining and Analysis, Beijing, China, 12–16 August 2012.
10. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; ACM: New York, NY, USA, 2014; pp. 701–710.
11. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. LINE: Large-scale Information Network Embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
12. Liu, L.; Cheung, W.K.; Li, X.; Liao, L. Aligning Users Across Social Networks Using Network Embedding. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 1774–1780.
13. Man, T.; Shen, H.; Liu, S.; Jin, X.; Cheng, X. Predict Anchor Links across Social Networks via an Embedding Approach. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 1823–1829.
14. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.

15. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 24–27 August 2016; pp. 855–864.
16. Yang, C.; Liu, Z.; Zhao, D.; Sun, M.; Chang, E.Y. Network representation learning with rich text information. In Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25 July–1 August 2015; pp. 2111–2117.
17. Pan, S.; Wu, J.; Zhu, X.; Zhang, C.; Wang, Y. Tri-party deep network representation. *Network* **2016**, *11*, 12.
18. Huang, X.; Li, J.; .; Hu, X. Label informed attributed network embedding. In Proceedings of the 10th ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 731–739.
19. Klau, G.W. A new graph-based method for pairwise global network alignment. *BMC Bioinform.* **2009**, *10*, 1–9. [CrossRef] [PubMed]
20. Foggia, P.; Gennaro, P.; Mario, V. Graph matching and learning in pattern recognition in the last 10 years. *Int. J. Pattern Recognit. Artif. Intell.* **2014**, *28*, 1450001. [CrossRef]
21. Melnik, S.; Garcia-Molina, H.; Rahm, E. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In Proceedings of the International Conference on Data Engineering, San Jose, CA, USA, 26 February–1 March 2002; pp. 117–128.
22. Kong, X.; Zhang, J.; Yu, P.S. Inferring anchor links across multiple heterogeneous social networks. In Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, San Francisco, CA, USA, 27 October–1 November 2013; pp. 179–188.
23. Zhang, S.; Tong, H. FINAL: Fast Attributed Network Alignment. In Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 24–27 August 2016; pp. 1345–1354.
24. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the 19th International Symposium on Computational Statistics, Paris, France, 22–27 August 2010; Springer: New York, NY, USA, 2010; pp. 177–186.
25. Zhang, Y.; Tang, J.; Yang, Z.; Pei, J.; Yu, P.S. COSNET: Connecting heterogeneous social networks with local and global consistency. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1485–1494.
26. Zhong, E.; Fan, W.; Wang, J.; Xiao, L.; Li, Y. Comsoc: Adaptive transfer of user behaviors over composite social network. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 696–704.
27. Mu, X.; Zhu, F.; Wang, J.; Wang, J.; Wang, J.; Zhou, Z.H. User Identity Linkage by Latent User Space Modelling. In Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 24–27 August 2016; pp. 1775–1784.