Link Prediction in Aligned Heterogeneous Networks

Fangbing Liu[1,2] and Shu-Tao Xia[1,2(✉)]

[1] Graduate School at Shenzhen, Tsinghua University, Beijing, China
[2] Tsinghua National Laboratory for Information Science and Technology, Beijing, China
lfb13@mails.tsinghua.edu.cn, xiast@sz.tsinghua.edu.cn

Abstract. Social networks develop rapidly and often contain heterogeneous information. When users join a new social network, recommendation affects their first impressions on this social network. Therefore link prediction for new users is significant. However, due to the lack of sufficient active data of new users in the new social network (target network), link prediction often encounters the cold start problem. In this paper, we attempt to solve the user-user link prediction problem for new users by utilizing data in a similar social network (source network). In order to bridge the two networks, three categories of local features related to single edge and one category of global features associated with multiple edges are selected. The Aligned Factor Graph (AFG) model is proposed for prediction, and *Aligned Structure Algorithm* is used to reduce the factor graph scale and keep the prediction performance at the same time. Experiments on two real social networks, i.e., Twitter and Foursquare show that AFG model works well when users leave little data in target network.

Keywords: Link prediction · Heterogeneous network · Aligned factor graph model

1 Introduction

In recent years, Social networks have become part of our life. When users join a new social network, their first impressions are very important to keep them active in this network. Thus how to predict future links for new users according to the current snapshot of the network is significant.

Link prediction can be seen as a classification problem. A classifier trained with simple topology features such as the number of common neighbors and the Adamic/Adar measure can successfully identify missing links in social networks [1]. Weak ties and interaction activities can also be useful for inference [2,3].

This research is supported in part by the Major State Basic Research Development Program of China (973 Program, 2012CB315803), the National Natural Science Foundation of China (61371078), and the Research Fund for the Doctoral Program of Higher Education of China (20130002110051).

T. Cao et al. (Eds.): PAKDD 2015, Part I, LNAI 9077, pp. 33–44, 2015.
DOI: 10.1007/978-3-319-18038-0_3

Actually, nodes in a social network often have abundant attributes such as time and location [4]. In addition, geographic distance has been shown to play an important role in creating new social connections [5]. In [6,7], network structure and node attributes are used simultaneously to improve prediction performance.

Many of the current studies mainly focus on a single social network. However, sometimes data in one network is not sufficient to train a good classifier. In particular, when users join a new network (target network), link prediction will encounter the cold start problem [8]. But if we can use data from another network (source network), the prediction performance should be better intuitively. In general, there are two ways to utilize the source network to help prediction, one is based on transfer learning through different feature spaces and the other is based on the factor graph. In the transfer learning method, items having both features in source space and target space are utilized [9]. The factor graph method uses the phenomenon that different social networks obey common rules such as triad social balance and triad status balance [10,11].

The works [9–11] focus on information transfer between *two different types of networks*. A widespread phenomenon is that some social networks are similar to each other except for some specific services. Users often have accounts in multiple social networks to enjoy distinctive services. Networks connected by accounts of same users are aligned networks. Link prediction for new users in aligned networks is first discussed in [12]. Though rich features are used for training, the important fact that user-user relationships affect each other is ignored.

In this paper, we study the link prediction problem for new users in target network from a new perspective. And the aligned source network is utilized to solve the cold start problem. Our method can get good prediction performance estimated by Area Under Curve (Auc) and Accuracy (Acc). The contributions can be summarized as follows:

- Three categories of local features and one category of global features are selected, which describe the social networks accurately and reflect the edges interaction. These features play an important role in improving prediction performance.
- An Aligned Factor Graph (AFG) model is proposed to solve the link prediction problem for new users in target network, making full use of a similar source network. It performs well when we encounter the cold start situation. In addition, in order to control the scale of the factor graph and guarantee an efficient inference, *Aligned Sturcture Algorithm* is used in building model.
- Experiments on two real social networks - Twitter and Foursquare are carried out and results show that AFG model improves prediction performance by utilizing source network data when compared with the Basic Factor Graph (BFG) model. And AFG model performs better than *SCAN-PS* model [12].

This paper is organized as follows: Section 2 gives basic definitions and related works in link prediction. Meanwhile, BFG model is introduced. Section 3 is our prediction method. Aligned Factor Graph (AFG) model is proposed and the *Aligned Structure Algorithm* is demonstrated. Besides, the parameter learning algorithm

and feature selection are discussed. Section 4 includes some experimental results and analysis. Section 5 is the conclusion.

2 Preliminaries and Related Works

Definition 1. *(Aligned Heterogeneous Networks [12]): Let V_i^k be the set of the same kind of nodes in network G_k, $V_k = \cup_i V_i^k$ is the set of different kinds of nodes. $E_k = \cup_i E_i^k$ is the set of different kinds of edges. Let f be a one-to-one mapping between user $u_i^s \in U_s \subseteq V_s$ in the source network and user $u_j^t \in U_t \subseteq V_t$ in the target network, if $\exists(F = \cup_{i,j} f(u_i^s, u_j^t)) \neq \emptyset$, then network $G_s = (V_s, E_s)$, $G_t = (V_t, E_t)$ are called aligned heterogeneous networks. The link (u_i^s, u_j^t) is called an anchor link and all these links form the set of anchor links E_A.*

Definition 2. *(Edge Descriptor): An edge e_{ij} can be described as $d_{ij} = (l_{ij}, p_{ij})$, where l_{ij} is the edge label belonging to $\{0, 1\}$, p_{ij} is the probability that e_{ij} having this label. $l_{ij} = 0$ means the edge does not exist.*

Definition 3. *(Triad Social Balance [14]): Undirected edges between three users form a triad. It is social balanced if three or one edge exists.*

Definition 4. *(Triad Status Balance [15]): Directed edges between three users form a triad. It is status balanced if three edges are not in a directed cycle.*

There are many works which use source network to help prediction in the target network. In [16], relationship prediction is studied under space feature transfer learning framework and inter-domain edges are enhanced by discovering new edges and strengthening existing ones. In [17,18], domain connection sparsity and data non-consistent problem are studied .

The prediction method based on factor graph concern triad features transfer between two different networks [10] and the *BFG* model [11] is used. Friend recommendation problem is solved by limiting friend candidates in two hops to keep factor graph in bearable scale [11]. And parameters of triad features are the same in source and target networks during training.

A factor graph [19] is defined as a bipartite graph containing variable nodes and factor nodes. In the *BFG* model, the user-user relationship e_{ij} between user u_i and u_j is mapped to a variable node v_{ij} in the factor graph, while variable nodes connecting to the same factor node reflect the interactive influence between relationships' formation. A simple explanation for *BFG* model is shown in Fig. 1.

3 Social Network Prediction

We try to solve the link prediction problem for new users in target network by utilizing data of aligned heterogeneous source network. Firstly, we extend the *BFG* model to the Aligned Factor Graph (*AFG*) model. Besides, the *Aligned Structure Algorithm* is used for controlling factor graph scale when building the model. Secondly, the parameter inference framework is proposed. Thirdly, a detailed parameter learning algorithm is studied. Fourthly, new user links are inferred by maximizing an objective function. At last, both local and global features used in prediction are given.

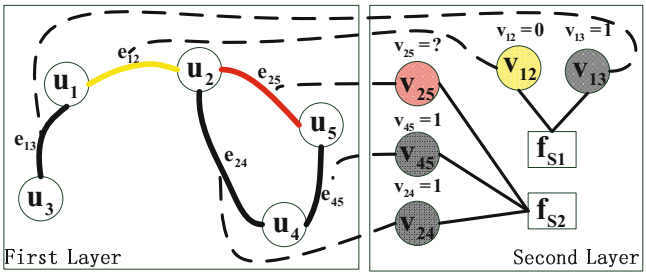

Fig. 1. BFG model. The first layer is observations, and the second layer is a factor graph. Each observation corresponds to a variable node. Black edge in the first layer means friend relationship exists between two users and the corresponding variable nodes's state is 1. Yellow edge indicates no friendship and variable node state is 0. Red edge represents unobserved relationship and variable node's state is ?, i.e., unknown.

3.1 The Aligned Factor Graph Model

The *AFG* model is also a two layer model. The first layer is composed of two observations deriving from source network G_s and target network G_t. The second layer is a factor graph containing two aligned parts $FG = \{FG_s \cup FG_t\}$. A more intuitive description of *AFG* model is shown in Fig. 2. The two networks in the first layer are fully aligned networks. Relationships between each pair of users are taken into consideration. Thus we can also find one-to-one mappings between variable nodes in FG_s and FG_t. Moreover, if variable node v_j^t's state is unknown in FG_t, the structure of v_j^t must be the same with the structure of the corresponding variable node v_i^s in FG_s. Local features belonging to variable nodes in the second layer can be got according to the corresponding edges' attributes in the first layer. Global features belonging to factor nodes are drawn from the edge cycles in the first layer, determining the factor graph structure.

Building *AFG* model efficiently is important for prediction. Firstly, the first layer observations can be got easily given G_s and G_t. Secondly, states of all variable nodes in the second layer are determined according to the observations. $state = 1$ and $state = 0$ variable nodes are state-known variable nodes while $state =?$ variable nodes belong to the state-unknown set. Thirdly, for combinations of state-known variable nodes satisfying global features defined in section 3.5, we build a factor node and connect it with the variable nodes in this combination. Fourthly, take the state-unknown variable nodes into consideration. If we build a factor node for each combination of variable nodes, the factor graph scale will be too large and the complexity will be too high. However, if we build a factor node and connect it with variable nodes randomly, the prediction performance will decrease. In this paper, Algorithm 1 is used to determine the accurate structures of state-unknown variable nodes.

3.2 Parameters Inference Framework

The first layer can be built given source network $G_s = (U_s, E_s, A_s)$ and target network $G_t = (U_t, E_t, A_t)$, where U_s, U_t are the sets of users, E_s, E_t are the sets

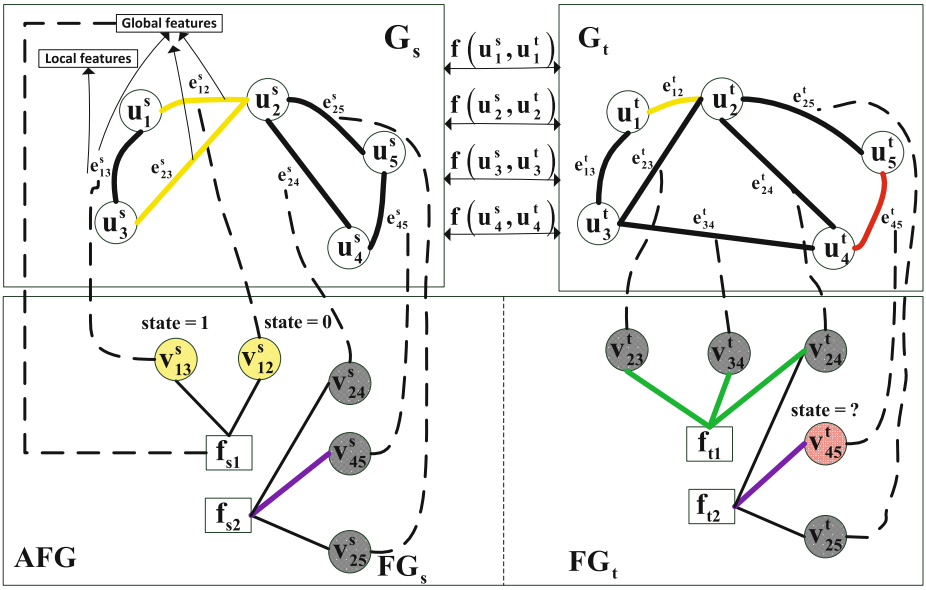

Fig. 2. AFG model. The second layer has an aligned structure. Edge color meanings are the same as Fig. 1. Red variable node v^t_{45} is connected to the factor node f_{t2} to keep aligned structure with FG_s as purple lines show. However, v^t_{23},v^t_{34},v^t_{24} can have different structures from FG_s, shown in green lines, because their states are known. v^s_{23},v^s_{34},v^s_{24} are not connected with a factor node in FG_s.

of user-user relationships, A_s, A_t are the sets of local attribute vectors belonging to edges. According to the observations in the first layer, a factor graph $FG = \{FG_s, FG_t\} = \{V_s, F_s, EF_s, V_t, F_t, EF_t\}$ in the second layer can be established, where V_s, V_t are the sets of variable nodes, F_s, F_t are the sets of factor nodes and EF_s, EF_t are the edge sets. In network G_t (so does G_s), each $e^t_{ij} \in E_t$ is associated with an attribute vector $a^t_{ij} \in A_t$ and is mapped to a variable node $v^t_{ij} \in FG_t$. e^t_{ij} has an edge descriptor $d^t_{ij} = (l^t_{ij}, p^t_{ij})$ related to v_{ij}'s state and marginal probability. As all edge descriptors in G_s are known while only part of edge descriptors in G_t are known, the link prediction problem can be described as maximizing the following probability

$$P(D_t, D_s|G_s, G_t) = \prod_{ij} f_l(v^s_{ij}, a^s_{ij}) g_c(v^s_{ij}, G(v^s_{ij})) \prod_{pq} f_l(v^t_{pq}, a^t_{pq}) g_c(v^t_{pq}, G(v^t_{pq})) \quad (1)$$

where G_t is the target network, G_s is the source network. D_t and D_s are the sets of edge descriptors in G_t, G_s.

The state of a variable node is affected by two features

- $f_l(v_{ij}, a_{ij})$: local feature, it describes how local attributes influence the friend relationship formation between user u_i and u_j.
- $g_c(v_{ij}, G(v_{ij}))$: global feature, it describes how two or three edges interact in forming the relationship. $G(v_{ij})$ is the set of variable nodes connecting to the same factor node with v_{ij}.

Algorithm 1. Aligned Structure Algorithm

Input: Source network G_s, Target network G_t
Output: $FG = \{FG_s \cup FG_t\}$

for all Combinations of state-unknown variable nodes (v_p^t, v_q^t) **do**
  Find v_p^t, v_q^t's one-to-one mapping variable nodes v_i^s, v_j^s in FG_s;
  if Combination (v_i^s, v_j^s) satisfies global features defined for two nodes **then**
    Build a factor node f_n^t and connect it with v_p^t, v_q^t;
  end if
end for
for all Combinations of state-unknown variable nodes (v_p^t, v_q^t, v_r^t) **do**
  Find v_p^t, v_q^t, v_r^t's one-to-one mapping variable nodes v_i^s, v_j^s, v_k^s in FG_s;
  if Combination (v_i^s, v_j^s, v_k^s) satisfies global features defined for three nodes **then**
    Build a factor node f_n^t and connect it with v_p^t, v_q^t, v_r^t;
  end if
end for

The two kinds of features can be instantiated using the *Markov Field* or the *Bayesian Theory*. In this paper, the *Hammersley-Clifford Theorem* [20] is used and the two probabilities are defined as

$$f_l(v_{ij}, a_{ij}) = \frac{1}{Z_1} \times \exp\{\sum_k \alpha_k r_k(a_{ij}^k)\} \tag{2}$$

$$g_c(v_{ij}, G(v_{ij})) = \frac{1}{Z_2} \times \exp\{\sum_c \sum_d \beta_d h_d(G(v_{ij}))\} \tag{3}$$

where Z_1, Z_2 are the normalization factors, k is the local attribute index, a_{ij}^k represents the k^{th} attribute in attribute vector a_{ij}. $G(v_{ij})$ is the set of variable nodes concerning v_{ij} and $|G(v_{ij})| = c$. If three edges affect each other, then $c = 3$. r_k is the k^{th} local feature function. For example, it can be a function calculating common neighbor number. h_d is the d^{th} global feature function. For instance, if a triad is social balanced, $h_d = 1$. α_k, β_d are the weights of features.

Then the joint probability defined by Eq. (1) can be written as

$$\begin{aligned} P(D_t, D_s|G_s, G_t) = \frac{1}{Z} \times \prod_{ij} \prod_{pq} \exp\{\sum_k \alpha_k (r_k(a_{pq}^{kt}) + \\ r_k(a_{ij}^{ks})) + \sum_c \sum_d \beta_d \{h_d(G(v_{pq}^t)) + h_d(G(v_{ij}^s))\}\} \end{aligned} \tag{4}$$

where Z is normalization factor. Thus, the source and target networks union objective function is

$$\begin{aligned} O(\theta) &= \log P(D_t, D_s|G_s, G_t) \\ &= \sum_k \alpha_k \{ \sum_{p=1}^{|U_{new}^t|} \sum_{q=1}^{|U_{all}^t|} (r_k(a_{pq}^{kt})) + \sum_{m=1}^{|U_{new}^s|} \sum_{n=1}^{|U_{all}^s|} (r_k(a_{mn}^{ks}))\} \\ &+ \sum_c \sum_d \beta_d \{ \sum_{p=1}^{|U_{new}^t|} \sum_{q=1}^{|U_{all}^t|} h_d(G(v_{pq}^t)) + \sum_{m=1}^{|U_{new}^s|} \sum_{n=1}^{|U_{all}^s|} h_d(G(v_{mn}^s))\} - \log Z \end{aligned} \tag{5}$$

Algorithm 2. Learning Algorithm

Input: *Learning Rate* η
Output: *Model Parameters* θ

1: **repeat**
2: Calculate $E_{p(D_{tu}|D_s,D_{tl},G_s,G_t)}(r_k(a_n^{st}))$, $E_{p(D_{tu}|D_{tl},D_s,G_s,G_t)}(h_d(G(v_m^{st})))$ using the LBP algorithm;
3: Calculate $E_{p(D_s,D_t|G_s,G_t)}(r_k(a_n^{st}))$, $E_{p(D_t,D_s|G_s,G_t)}(h_d(G(v_m^{st})))$ using the LBP algorithm;
4: Calculate gradient according to Eqs. (6) and (7);
5: Update parameter set θ with learning rate
6: $\theta_{new} = \theta_{old} - \eta \times \frac{\partial O(\theta)}{\partial \theta}$
7: **until** Converage
8: Output θ

where U_{new}^t is the set of new users, U_{all}^t is the set of all users in G_t (so does G_s). We try to find parameter set $\theta = (\alpha, \beta)$ that maximizing the objective function.

3.3 Learning Algorithm

In order to solve the objective function, the gradient decent algorithm is used. As Z is the normalization factor, all variable nodes' likelihoods in the factor graph need to be calculated including the state-unknown variable nodes. The gradients of parameters are calculated as follows

$$\frac{\partial O(\theta)}{\partial \alpha_k} = E_{p(D_{tu}|D_{tl},D_s,G_s,G_t)}(r_k(a_n^{st})) - E_{p(D_s,D_t|G_s,G_t)}(r_k(a_n^{st})) \tag{6}$$

$$\frac{\partial O(\theta)}{\partial \beta_d} = E_{p(D_{tu}|D_{tl},D_s,G_s,G_t)}\{h_d(G(v_m^{st}))\} - E_{p(D_t,D_s|G_s,G_t)}\{h_d(G(v_m^{st}))\} \tag{7}$$

where a_n^{st} is local attribute vector associating with variable node v_n^{st} in AFG model's second layer and v_m^{st} is the m^{th} variable node. D_{tu} is the set of unknown descriptors and D_{tl} is the set of known descriptors. $E_{p(D_{tu}|D_{tl},D_s,G_s,G_t)}(r_k(a_n^{st}))$ is the expectation of the local function given all known descriptors of edges, while $E_{p(D_s,D_t|G_s,G_t)}(r_k(a_n^{st}))$ is the expectation given the estimated model. As the factor graph has different topology, it is hard to directly calculate the second part. In this paper, we use Loopy Belief Propagation (LBP) [21] to approximate the gradients. With LBP, the marginal probilities of different states of variable nodes can be calculated. After this, we sum over all nodes to obtain the gradient. The detailed algorithm is shown in Algorithm 2.

3.4 New User Link Inference

Model parameters θ can be got through learning. Then new user link inference problem is defined as finding the descriptors that maximizing the probability

$$O(D_{tu}) = P(D_{tu}|D_{tl}, D_s, G_s, G_t, \theta) \tag{8}$$

where D_{tu} is the set of unknown edge descriptors, D_{tl} is the set of known edge descriptors. The LBP algorithm is used to compute the marginal probability of each variable node v_i^{st} in factor graph. And we choose the state $l_i^{st} \in \{0, 1\}$ with larger marginal probability $p_i^{st} = \max\{p(0|\theta), p(1|\theta)\}$ as v_i^{st}'s state. The edge descriptor correspinding to variable node v_i^{st} is $d_i^{st} = (l_i^{st}, p_i^{st})$.

Time cost is also very important when applying the prediction framework. If n users exist in social network, building AFG costs $\mathcal{O}(n^3)$ time. Parameter learning complexity is $\mathcal{O}(cnt)$, t is the number of iterations, c is a constant. Thus, the whole prediction algorithm can be finished in polynomial time.

3.5 Feature Selection

Table 1 is a list of all local features. As the networks we study are heterogenous and contain different types of data, three categories of local features can be selected, namely topology feature, location feature and time feature. s stands for source user and t stands for target user of an edge. FI_t is the set of users who follow user t and F_t is the set of users whom user t follows. Loc_t is location vector of user t, each element is the user's visited number of this location. Tim_t is time vector of user t with length 24, corresponding to the 24 hours of a day.

Taking the interactive effects of edges into consideration, one category of global features is drawn. We find that more than 90% triads in our data set are triad social balanced and triad status balanced. According to this observation, we choose the global features in Table 2.

Table 1. Local features

Category	*Feature Name*	*Definition*
Topology	BoolOpinionLeader	0 *or* 1
	InDegree	$\|FI_s\|, \|FI_t\|$
	OutDegree	$\|F_s\|, \|F_t\|$
	TotalDegree	$\|FI_s \cup F_s\|, \|FI_t \cup F_t\|$
	NumCommonNeighbor	$\|(FI_s \cup F_s) \cap (FI_t \cup F_t)\|$
	NumTotalNeighbor	$\|(FI_s \cup F_s) \cup (FI_t \cup F_t)\|$
	SimAdamic	$\sum_{i \in (FI_s \cup F_s) \cap (FI_t \cup F_t)} \{1/\log \|FI_i \cup F_i\|\}$
	SimJaccard	$\|(FI_s \cup F_s) \cap (FI_t \cup F_t)\| / \|(FI_s \cup F_s) \cup (FI_t \cup F_t)\|$
Location	LocationDis	$(\sum_i (Loc_{s,i} - Loc_{t,i})^2)^{1/2}$
	LocationCosine	$(Loc_s \cdot Loc_t)/(\parallel Loc_s \parallel \cdot \parallel Loc_t \parallel)$
	LocationJaccard	$\|Loc_s \cap Loc_t\| / \|Loc_s \cup Loc_t\|$
Time	TimeDis	$(\sum_i (Tim_{s,i} - Tim_{t,i})^2)^{1/2}$
	TimeCosine	$(Tim_s \cdot Tim_t)/(\parallel Tim_s \parallel \cdot \parallel Tim_t \parallel)$
	TimeExtendJaccard	$(Tim_s \cdot Tim_t)/(\|Tim_s\|^2 + \|Tim_t\|^2 - Tim_s \cdot Tim_t)$

Table 2. Global features

Feature Name	*Definition*
CommonSourceUser	*0 or 1*
CommonTargetUser	*0 or 1*
SocialBalance	*0 or 1*
SocialStatus	*0 or 1*

4 Experiment

4.1 Experiment Settings and Results

We use Twitter and Foursquare data sets and the same method as [12] to divide data for 5-cross-validation. Firstly, we randomly choose 1000 users to form two fully aligned networks. Secondly, 20% of users are chosen as new users. Thirdly, all existing friend relationship edges related to new users are put into an existing link set, equivalent number of non-existing friend relationship edges are put into non-existing link set. Fourthly, both the existing link set and the non-existing link set are divided into five parts. Fifthly, if the old users' information is used, just keep balance when expanding the two link sets. The ratio of new users' data used for training is defined as user novelty. Ratio 0.0 means brand-new users. All relationships related to new users are sampled according to the setting novelty.

Table 3. Experiment settings and results

	Target	*Source*	*Model*	*Baseline*	*Auc* ↑	*Acc* ↑
Group1	Twitter	None	*BFG*	*TRAD*	−3%	2%
	Twitter	Foursquare	*AFG*	*SCAN-PS*	10%	11%
	Twitter	Foursquare	*AFG*	*BFG*	36%	31%
Group2	Foursquare	None	*BFG*	*TRAD*	−15%	−9%
	Foursquare	Twitter	*AFG*	*SCAN-PS*	7%	3%
	Foursquare	Twitter	*AFG*	*BFG*	32%	25%

Two groups of experiments are carried out in this paper. Traditional Link Prediction *(TRAD)* and Supervised Cross Aligned Networks Link Prediction with Personalized Sampling *(SCAN-PS)* proposed in [12] are used as baseline methods. *SCAN-PS* merges features extracted from the anchor link in source network to expand the feature vector of corresponding link in target network to train a classifier. *Auc* and *Acc* are the performance evaluation criteria. Detailed comparative models and main results are shown in Table 3.

4.2 Performance Analysis

Fig. 3 is the results of first group experiments. Twitter is the target network, Foursquare is the source network in this group experiments.

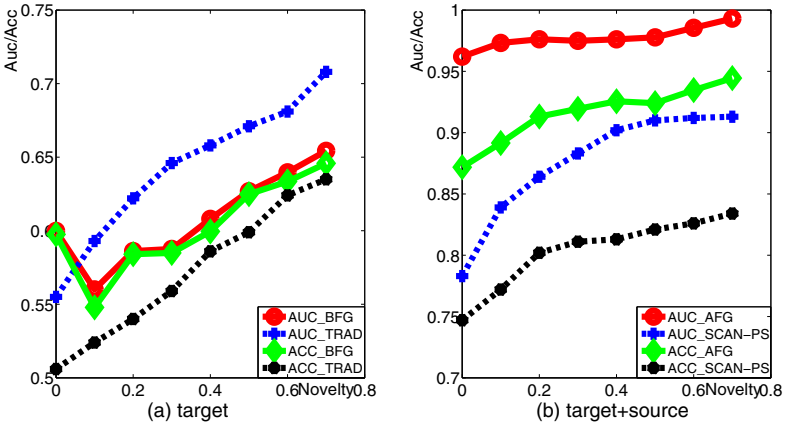

Fig. 3. Source network: Foursquare, target network: Twitter. It shows how *Auc* and *Acc* change with the user novelty. 1) In (*a*), *BFG* model has a higher *Acc* in average while *TRAD* model performs well under *Auc*. 2) In (*b*), *AFG* model improves *Auc* by about 10% and *Acc* by about 11% than *SCAN-PS* model. 3) *AFG* model in (*b*) improves *Auc* by 36% and *Acc* by 31% in average compared with *BFG* model in (*a*).

- In Fig. 3(*a*), *BFG* model performs worse than *TRAD* model in *Auc* because there are many state-unknown variable nodes in target network, the factor graph structure can not be decided accurately. Inaccurate factor graph structure decreases *BFG* model performance, but it has no effect on *TRAD* model, which only makes use of the local features. As Twitter is follow-follow network, users having most fans play important role in network formation. That is the reason why we get high *Auc* and *Acc* when user novelty is 0.0.
- In Fig. 3(*b*), *AFG* model performs better than *SCAN-PS* model both in *Auc* and *Acc*. That is because source network information expands the training set and *Aligned Sturcture Algorithm* determines accurate factor graph structure.
- *AFG* model uses Foursquare to help new user link prediction in Twitter while *BFG* model only use Twitter data. Comparing *AFG*' performance in Fig. 3(*b*) and *BFG*'s performance in Fig. 3(*a*), we find that *AFG* model can make full use of source network to improve the prediction performance.

Though target network and source network are similar, they also have own characteristics. Foursquare provides location based service while Twitter provides Tweet service. In order to prove that *AFG* model is suitable to solve new user link prediction problem in similar aligned networks regardless of their positions, we use Foursquare as target network and Twitter as source network in second group experiments. And the results of second group experiments are shown in Fig. 4.

- In Fig. 4(*a*), *BFG* model performs worse than *TRAD* model, keeping the same trend with Fig. 3(*a*) for the same reason.

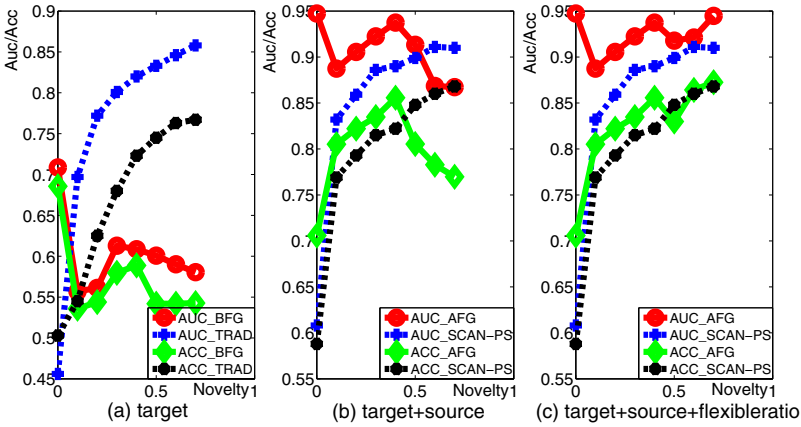

Fig. 4. Source network: Twitter, target network: Foursquare. It shows how *Auc* and *Acc* change with the user novelty. 1) In (*a*), *BFG* model performs worse than *TRAD* model. 2) In (*b*), *AFG* model performs better than *SCAN-PS* model when user novelty is less than 0.5. 3) In (*c*), *AFG* model improves the *Auc* by 7% and *Acc* by 3% in average than *SCAN-PS* model. 4) *AFG* model in (*c*) improves *Auc* by 32% and *Acc* by 25% in average compared with *BFG* model in (*a*).

- In Fig. 4(*b*), the *AFG* model performance increases gradually before user novelty reaches 0.5, then its performance decreases. That is because the Twitter part in union training set contains noise. Only 6.5% users in Twitter have location data [12]. We use the corresponding users' location data in Foursquare on condition that the user-location links are in the training set of Foursquare part. The location data replacement causes noise, though the training set is expanded.
- In Fig. 4(*c*), *AFG* model performs better than *SCAN-PS* model both in *Auc* and *Acc*. Balance between training data amount and low data noise is achieved by using different ratio of the source network data as user novelty changes. This method improves the performance compared with curves in Fig. 4(*b*) when user novelty exceeds 0.5.

5 Conclusion

The link prediction problem for new users is studied in this paper. Recommendations for new users have significant influence on their keeping active in this social network. However, the cold start problem is often encountered. The *AFG* model is proposed to utilize data from a similar source network to help prediction in target network. Three categories of local features and one category of global features are put forward for training. The *Aligned Structure Algorithm* is brought up to reduce the scale of the factor graph and keep high prediction accuracy when building the model. Experiments on Twitter and Foursquare show that *AFG* model can make full use of source network data to improve prediction

performance compared with *BFG* model, which can only use the target network data. And *AFG* model performs better than *SCAN-PS* model. *Auc* is increased by 10% and *Acc* is increased by 11% in average when Foursquare is source network and Twitter is target network. On the other hand, 7% *Auc* and 3% *Acc* improvements are achieved when swapping positions of the two networks.

References

1. Fire, M., Tenenboim, L., Lesser, O., Puzis, R., Rokach, L., Elovici, Y.: Link prediction in social networks using computationally efficient topological features. In: SocialCom, pp. 73–80 (2011)
2. Lü, L., Zhou, T.: Link prediction in weighted networks: The role of weak ties. EPL 89(1) (2010)
3. Xiang, R., Neville, J., Rogati, M.: Modeling relationship strength in online social networks. In: WWW, pp. 981–990 (2010)
4. Scellato, S., Noulas, A., Mascolo, C.: Exploiting place features in link prediction on location-based social networks. In: SIGKDD, pp. 1046–1054 (2011)
5. Allamanis, M., Scellato, S., Mascolo, C.: Evolution of a location-based online social network: analysis and models. In: IMC, pp. 145–158 (2012)
6. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: WSDM, pp. 635–644 (2011)
7. Davis, D., Lichtenwalter, R., Chawla, N.V.: Supervised methods for multi-relational link prediction. Social Network Analysis and Mining **3**(2), 127–141 (2013)
8. Leroy, V., Cambazoglu, B.B., Bonchi, F.: Cold start link prediction. In: SIGKDD, pp. 393–402 (2010)
9. Dai, W., Chen, Y., Xue, G.R., Yang, Q., Yu, Y.: Translated learning: Transfer learning across different feature spaces. In: NIPS, pp. 353–360 (2008)
10. Tang, J., Lou, T., Kleinberg, J.: Inferring social ties across heterogenous networks. In: WSDM, pp. 743–752 (2012)
11. Dong, Y., Tang, J., Wu, S., Tian, J., Chawla, N.V., Rao, J., Cao, H.: Link prediction and recommendation across heterogeneous social networks. In: ICDM, pp. 181–190 (2012)
12. Zhang, J., Kong, X., Yu, P.S.: Predicting social links for new users across aligned heterogeneous social networks. In: ICDM, pp. 1289–1294 (2013)
13. Liu, L., Tang, J., Han, J., Jiang, M., Yang, S.: Mining topic-level influence in heterogeneous networks. In: CIKM, pp. 199–208 (2010)
14. Zheng, X., Zeng, D., Wang, F.Y.: Social balance in signed networks. Information Systems Frontiers 1(19) (2014)
15. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: SIGCHI, pp. 1361–1370 (2010)
16. Cremonesi, P., Tripodi, A., Turrin, R.: Cross-domain recommender systems. In: ICDMW, pp. 496–503 (2011)
17. Tang, J., Wu, S., Sun, J., Su, H.: Cross-domain collaboration recommendation. In: SIGKDD, pp. 1285–1293 (2012)
18. Lu, Z., Pan, W., Xiang, E.W., Yang, Q., Zhao, L., Zhong, E.: Selective transfer learning for cross domain recommendation. In: SDM pp. 641–649 (2013)
19. Frey, B.J., Kschischang, F.R., Loeliger, H.A., Wiberg, N.: Factor graphs and algorithms. In: Allerton, pp. 666–680 (1997)
20. Hammersley, J.M., Clifford, P.: Markov fields on finite graphs and lattices (1971)
21. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Bethe free energy, Kikuchi approximations, and belief propagation algorithms. In: NIPS (2001)