University of Illinois at Chicago

PhD Qualifier Examination Paper

Link Prediction across Heterogeneous Social Networks: A Survey

---

Author:
Jiawei Zhang

Supervisor:
Dr. Philip S. Yu

Committee Members: Dr. Bing Liu (Chair)
Dr. Philip S. Yu
Dr. Ouri E. Wolfson

March 20, 2014

Abstract

Online social networks have gained great success in recent years. Some online social networks only involving users and social links among users can be represented as homogeneous networks. Meanwhile, some other social networks containing abundant information, which include multiple kinds of nodes and complex relationships, can be denoted as heterogeneous networks. Predicting the missing links or links that will be formed in the future based on a snapshot of social networks is formally defined as the link prediction problem. Link prediction problems have extensive applications in real-world social networks and many concrete social services can be cast as link prediction tasks, e.g., friend and location recommendations can all be solved as the problem of predicting social links among users and the location links between users and locations. Link prediction problems have been an important research topic for many years and a large number of different methods have been proposed so far.

This article summaries the existing link prediction methods for both homogeneous and heterogeneous networks, which include various unsupervised link predicators, random walk based link prediction methods, methods based on matrix factorization techniques, supervised link prediction methods and meta paths based link prediction methods. Meanwhile, as proposed in recent works, people are usually involved in multiple social networks simultaneously nowadays and networks sharing common users are formally defined as the aligned networks. In this article, we will also introduce the latest progress of link prediction problems across multiple aligned heterogeneous networks. The link prediction problems across aligned networks can include anchor link prediction problem and link transfer across aligned heterogeneous networks. We will introduce the newly proposed methods to solve these problems in details and, finally, we will conclude this survey with a discussion about the future link prediction research works.

Contents

Chapter 1

Introduction

Online social networks, such as Facebook, Twitter and Foursquare, have become more and more popular in recent years. Some social networks involving one single type of nodes and links can be represented as homogeneous networks, while some other social network containing abundant information about: who, where, when and what [30], can be denoted as heterogeneous networks. Information entities in online social networks can be represented as nodes and the relationships among the nodes can be denoted as *links*, e.g., social connections among users can be cast as social links [60, 61], location check-ins can be indicated as location links between users and locations [61].

However, in some cases, not all links in social networks are observable, which can be (1) hidden by the users to protect personal privacy [5, 32, 50]; (2) missing due to the mistakes happened in crawling, storage or transmission of the network data [13, 18]. In other cases, the social networks studied can be dynamic [9, 42] and links within the networks can evolve with time. Many links that are nonexistent in the network can appear in the future [33, 24]. Therefore, predicting the missing links in social networks or potential links that will exist in the future can be an interesting problem.

Link prediction has extensive applications in real-world social networks and many concrete social services can be cast as link prediction tasks, e.g., friend recommendation services can be solved by predicting the social links among users [43, 50], location recommendation services can be regarded as the location link prediction task [57, 12].

According to the heterogeneity of networks, link prediction problems in online social networks can divided into two categories: (1) link prediction problems in homogeneous networks [33], (2) link prediction problems in heterogeneous networks [54, 44, 57, 12]. Meanwhile, according to the number of link types to be predicted, link prediction problems can be partitioned into two subsets: (1) single link prediction task [60, 33, 54, 44, 3, 39, 24], e.g., social link prediction or co-author link prediction, (2) collective link prediction task [61, 40, 53, 8, 15], which aims at predicting multiple kinds of links, e.g., social and location links, simultaneously. For each specific link prediction problem, many different link prediction approaches have been proposed, e.g., massive unsupervised predicators based on social similarity measures [33], methods based on random walk [22, 19, 31, 6, 47], methods based on matrix factorization [1, 46, 17], meta path based supervised link prediction methods [58, 44] and collective link prediction framework [61, 15].

In recent years, link prediction problems have many new developments. As proposed in [30, 60, 61], nowadays, to enjoy more online social services, people are usually getting involved in multiple different social networks simultaneously [30]. For example, people can participate in Foursquare to share reviews or tips about different locations or places with their friends. At the same time, they may use Twitter to post comments on the latest news, and turn to Facebook to share photos with relatives. These social networks sharing common users are formally defined as *multiple aligned networks*, which is first proposed in [30].

These shared users in different social networks are formally defined as the *anchor users* [30, 60, 61] as they can act like anchors fixing the networks they participate in, while the remaining unshared users are named as the *non-anchor users*. To represent the connections between aligned networks, the links between accounts of anchor users in different networks are defined as the *anchor links*, which is a new type of links first proposed in [30].

Across the aligned networks, many novel link prediction problems have been proposed so far, which include (1) anchor link prediction [30], which aims at predicting the anchor links between networks; (2) social link prediction for new users [60], which focuses on prediction social links for new users with information across aligned networks and can overcome the cold start problem; (3) collective social and location link prediction [61], which can predict the social links and location links across networks simultaneously.

In this article, we present a survey about both traditional and newly proposed link prediction problems and approaches. The article is organized as follows: we will introduce the definition of some important concepts, the formulation of problems and evaluation metrics in Chapter 2; link prediction problems and methods for homogeneous networks will be given in Chapter 3; we will describe the link prediction problems and methods for heterogeneous networks in Chapter 4; we will talk about newly introduced link prediction problems across aligned heterogeneous networks as well as the methods proposed to solve these problems in details in Chapter 5. Finally, we will conclude the article with future works in link prediction in Chapter 6.

Chapter 2

Problem Formulation

2.1 Terminology Definition

Definition 1 (Homogeneous Social Network): For a given social network $G = (V, E)$, where V is the node set and E is the link set. If all nodes in V are identical and all links in E are of the same type, then G is defined to be a *homogeneous social network*.
Definition 2 (Heterogeneous Social Network): A social network is *heterogeneous* if it contains multiple kinds of nodes and links. *Heterogeneous social networks* can be represented as $G = (V, E)$, where $V = \bigcup_i V_i$ is the union of different node sets and $E = \bigcup_i E_i$ is the union of heterogeneous link sets.
Definition 3 (Aligned Heterogeneous Social Networks): If two different social networks share some common users, then these two networks are called *aligned networks*. *Multiple aligned heterogeneous social networks* can be formulated as $\mathcal{G} = ((G^1, G^2, \cdots, G^n), (A^{1,2}, A^{1,3}, \cdots, A^{1,n}, A^{2,1}, \cdots, A^{n,(n-1)}))$, where $G^i, i \in \{1, 2, \cdots, n\}$ is a *heterogeneous social network* and $A^{i,j} \neq \emptyset, i, j \in \{1, 2, \cdots, n\}$ is the set of directed *anchor links* from G^i to G^j [30, 60, 61].
Definition 4 (Anchor Links): Let U^i and U^j be the user sets of G^i and G^j respectively. Link (u^i, v^j) is a directed *anchor link* from G^i to G^j iff. $(u^i \in U^i) \wedge (v^j \in U^j) \wedge (u^i$ and v^j are the accounts of the same user in G^i and G^j respectively) [30, 60, 61].

2.2 Link Prediction Problem Formulation

Let $G = (V, E)$ be the given network, where $V = \bigcup_i V_i$ is the union of various kinds of node sets in G and $E = \bigcup_i E_i$ is the union of link sets among these nodes in the network. From network G, a set of existing links $\mathcal{E}$ and a set of potential links to be predicted $\mathcal{L}$ can be extracted.

Traditional social link prediction models formulate the problem either as a label prediction problem, where existent and nonexistent links are labeled

Table 2.1: Confusion matrix of link prediction results.

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

as positive and negative links respectively, or as a existence probability estimation problems, where links predicted to be existent can have higher existence probabilities. Conventional methods aim at obtaining a link prediction model, M, built with links in $\mathcal{E}$ and apply the model to the potential social link set $\mathcal{L}$ to predict their labels and their existence probabilities. In other words, social link prediction model M can map links in $\mathcal{L}$ to their labels in {1, -1}, $f_M : \mathcal{L} \to \{1, -1\}$, where if link $l \in \mathcal{L}$ is predicted to be existent, then $f_M(l) = 1$; otherwise, $f_M(l) = -1$, or try to predict their existence probabilities (or confidence scores) in $[0, 1]$, $g_M : \mathcal{L} \to [0, 1]$.

2.3 Evaluation Metrics

For the prediction results, different evaluation metrics can be applied to measure the performance of model M. Considering, for example, based on the given link prediction results shown in confusion matrix (Table 2.1), the metrics that can evaluate the performance of model M include:

Evaluation Metrics for Methods with Labels Output

- *Accuracy*: $Accuracy = \frac{TP+TN}{TP+FN+FP+TN}$, which is the number of correctly classified instances in the test set divided by the total number of instances.
- *Precision*: $Precision = \frac{TP}{TP+FP}$, which is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.
- *Recall*: $Recall = \frac{TP}{TP+FN}$, which is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.
- *F1-Score*: $F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$, which is the harmonic mean of precision and recall.

Evaluation Metrics for Methods with Score Output

- *ROC Curve*: ROC curve is a plot of the *true positive rate* (tpr) against the *false positive rate* (fpr), where $tpr = \frac{TP}{TP+FN}$ and $fpr = \frac{FP}{TN+FP}$.
- *AUC*: AUC denotes the area under the ROC curve. Larger AUC corresponds to better classification results.

Chapter 3

Link Prediction for Homogeneous Networks

In this chapter, we will introduce the link prediction methods for homogeneous networks, e.g., $G = (V, E)$ containing users and social links among users. Existing link prediction methods for homogeneous networks can include massive unsupervised link predicators [33], random walk based link prediction method [22, 19, 31, 6, 47] and methods based on matrix factorization [1, 46, 17], etc.

3.1 Unsupervised Link Predicators

Traditional unsupervised link predicators can be divided into two main categories: (1) local neighbor based predicators and (2) global path based predicators.

3.1.1 Local Neighbor based Predicators

Local neighbor based predicators are based on local social information, i.e., neighbors of users in the network. Consider, for example, given a social link (u, v) in network G, where u and v are both users, neighbor sets of u, v can be represented as $\Gamma(u)$ and $\Gamma(v)$ respectively. Based on $\Gamma(u)$ and $\Gamma(v)$, we can obtain the following predicators measure the proximity of user u and v in network G.

1. *Preferential Attachment Index* (PA) [7]:

$$PA(u, v) = |\Gamma(u)| \, |\Gamma(v)| \, .$$

   $PA(u, v)$ uses the product of the degrees of users u and v in the network as the proximity measure, considering that new links are more likely to appear between users who have large number of social connections.

2. *Common Neighbor* (CN) [25]:

$$CN(u,v) = |\Gamma(u) \cap \Gamma(v)|\,.$$

$CN(u,v)$ uses the number of shared neighbor as the proximity score of user u and v. The larger $CN(u,v)$ is, the closer user u and v are in the network.

3. *Jaccard's Coefficient* (JC) [25]:

$$JC(u,v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}.$$

$JC(u,v)$ takes the total number of neighbors of u and v into account, considering that $CN(u,v)$ can be very large because each one has a lot of neighbors rather than they are strongly related to each other.

4. *Adamic/Adar Index* (AA) [2]:

$$AA(u,v) = \sum_{w \in (\Gamma(u) \cap \Gamma(v))} \frac{1}{\log|\Gamma(w)|}.$$

Different from $JC(u,v)$, $AA(u,v)$ further gives each common neighbor of user u and v a weight, $\frac{1}{\log|\Gamma(w)|}$, to denote its importance.

5. *Resource Allocation Index* (RA) [62]:

$$RA(u,v) = \sum_{w \in (\Gamma(u) \cap \Gamma(v))} \frac{1}{|\Gamma(w)|}.$$

$RA(u,v)$ gives each common neighbor a weight $\frac{1}{|\Gamma(w)|}$ to represent its importance.

All these predicators are called *local neighbor based predicators* as they are all based on users' local social information.

3.1.2 Global Path based Predicators

In addition to the local neighbor based predicators, many other predicators based on paths in the network have also been proposed to measure the proximity among users.

1. *Shortest Path* (SP) [24]:

$$SP(u,v) = \min\{|p_{u \rightsquigarrow v}|\},$$

where $p_{u \rightsquigarrow v}$ denotes a path from u to v in the network and $|p|$ represents the length of path p.

2. *Katz* [29]:

$$Katz(u,v) = \sum_{l=1}^{\infty} \beta^l \left| p^l_{u \rightsquigarrow v} \right|,$$

where $p^l_{u \rightsquigarrow v}$ is the set of paths of length l from u to v and parameter $\beta \in [0,1]$ is a regularizer of the predicator. Normally, a small β favors shorter paths as β^l can decay very quickly when β is small, in which case $Katz(u,v)$ will be behave like the predicators based on local neighbors.

3.2 Random Walk based Link Prediction

In addition to the unsupervised link predicators which can be obtained from the networks directly, there exists another category link prediction methods which can calculate the proximity scores among users based on *random walk* [22, 19, 31, 6, 47, 38, 25]. In this part, we will introduce the concept of random walk at first. Next, we will introduce the proximity measures based on random walk, which include the *commute time* [19, 38, 25], *hitting time* [19, 38, 25] and *cosine similarity* [19, 38, 25].

3.2.1 Random Walk

Let matrix $\mathbf{A}$ be the adjacency matrix of network G, where $A_{i,j} = 1$ iff. social link $(u_i, u_j) \in E$, where $u_i, u_j \in V$. The normalized matrix of $\mathbf{A}$ by rows will be $\mathbf{P} = \mathbf{D}_A^{-1}\mathbf{A}$, where diagonal matrix $\mathbf{D}_A$ of $\mathbf{A}$ has value $(D_A)_{i,i} = \sum_j A_{i,j}$ on its diagonal and $P_{i,j}$ stores the probability of stepping on node $u_j \in \mathcal{U}$ from node $u_i \in \mathcal{U}$. Let entries in vector $\mathbf{x}^{(\tau)}(i)$ denote the probabilities that a random walker is at user node $u_i \in V$ at time τ. Then [19, 38, 25],

$$\mathbf{x}^{(\tau+1)}(i) = \sum_j \mathbf{x}^{(\tau)}(j)\mathbf{P}_{j,i}.$$

In other words, the updating equation of vector $\mathbf{x}$ will be as follows:

$$\mathbf{x}^{(\tau+1)} = \mathbf{P}\mathbf{x}^{(\tau)}.$$

Keep updating $\mathbf{x}$ according to the following equation until convergence,

$$\begin{cases} \mathbf{x}^{(\tau+1)} = \mathbf{P}^T\mathbf{x}^{(\tau)}, \\ \mathbf{x}^{(\tau+1)} = \mathbf{x}^{(\tau)}. \end{cases}$$

We can obtain the final stationary distribution vector $\mathbf{v}$ to be:

$$\mathbf{v} = \mathbf{P}^T\mathbf{v}.$$

The above equation denotes that the final stationary distribution vector $\mathbf{v}$ is actually a eigenvector of matrix $\mathbf{P}^T$ corresponding to eigenvalue 1. Some existing works have pointed out that if a markov chain is *irreducible* [19] and

aperiodic [19] then the largest eigenvalue of the transition matrix will be equal to 1 and all the other eigenvalues will be strictly less than 1. In addition, in such condition, there will exist a unique stationary distribution which is vector $\mathbf{v}$ obtained at convergence of the updating equations.
Definition 5 (Irreducible): Network G is *irreducible* if there exists a path from every node to every other nodes in G [19].
Definition 6 (Aperiodic): Network G is *aperiodic* if the greatest common divisor of the lengths of its cycles in G is 1, where the greatest common divisor is also called the *period* of G [19].

3.2.2 Proximity Measures based on Random Walk

1. *Hitting Time* (HT) [19, 38, 25]:

$$HT(u,v) = \mathbb{E}\left(\min\{\tau \in \mathbb{N}^+, X^{(\tau)} = v | X^0 = u\}\right),$$

where variable $X^{(\tau)} = v$ denotes that a random walker is at node v at time τ.

$HT(u,v)$ counts the average steps that a random walker takes to reach node v from node u. According to the definition, the hitting time measure is usually asymmetric, $HT(u,v) \neq HT(v,u)$. Based on matrix $\mathbf{P}$ defined before, the definition of $HT(u,v)$ can be redefined as [19]:

$$HT(u,v) = 1 + \sum_{w \in \Gamma(u)} P_{u,w} HT(w,v).$$

2. *Commute Time* (CT) [19, 38]:

$$CT(u,v) = HT(u,v) + HT(v,u).$$

$CT(u,v)$ counts the expectation of steps used to reach node u from v and those needed to reach node v from u. According to existing works, the commute time, $CT(u,v)$, can be obtained as follows

$$CT(u,v) = 2m(L^{\dagger}_{u,u} + L^{\dagger}_{v,v} - 2L^{\dagger}_{u,v}).$$

where $\mathbf{L}^{\dagger}$ is the pseudo-inverse of matrix $\mathbf{L} = \mathbf{D}_A - \mathbf{A}$.

3. *Cosine Similarity based on* $\mathbf{L}^{\dagger}$ (CS) [19, 38]:

$$CS(u,v) = \frac{\mathbf{x}_u^T \mathbf{x}_v}{\sqrt{(\mathbf{x}_u^T \mathbf{x}_u)(\mathbf{x}_v^T \mathbf{x}_v)}}.$$

where, $\mathbf{x}_u = (\mathbf{L}^{\dagger})^{\frac{1}{2}} \mathbf{e}_u$ and vector $\mathbf{e}_u$ is a vector of 0s except the entries corresponding to node u that is filled with 1. According to existing works

[19, 38], the cosine similarity based on $\mathbf{L}^{\dagger}$, $CS(u,v)$, can be obtained as follows,

$$CS(u,v) = \frac{L^{\dagger}_{u,v}}{\sqrt{L^{\dagger}_{u,u} L^{\dagger}_{v,v}}}.$$

4. *Random Walk with Restart* (RWR) [19, 38, 25]: Based on the definition of random walk, if the walker is allowed to return to the starting point with a probability of $1-c$, where $c \in [0,1]$, then the new random walk method is formally defined as *random walk with restart*, whose updating equation is shown as follows:

$$\begin{cases} \mathbf{x}_u^{(\tau+1)} = c\mathbf{P}^T \mathbf{x}_u^{(\tau)} + (1-c)\mathbf{e}_u, \\ \mathbf{x}_u^{(\tau+1)} = \mathbf{x}_u^{(\tau)}. \end{cases}$$

Keep updating $\mathbf{x}$ until convergence, the stationary distribution vector $\mathbf{x}$ can meet

$$\mathbf{x}_u = (1-c)(\mathbf{I} - c\mathbf{P}^T)^{-1}\mathbf{e}_u.$$

The proximity measure based on random walk with restart between user u and v will be [19, 38, 25]

$$RWR(u,v) = \mathbf{x}_u(v),$$

where $\mathbf{x}_u(v)$ denotes the entry corresponding to v in vector $\mathbf{x}_u$.

3.3 Matrix Factorization based Link Prediction

Besides unsupervised link predicators and proximity measures based on random walk, many other methods based on matrix factorization have also been proposed to predict links in homogeneous networks [1, 46, 17].

Given the adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$ of network G, we propose to use a low-rank compact representation, $\mathbf{U} \in \mathbb{R}^{n \times d}, d < n$, to store social information for each user in the network. Matrix $\mathbf{U}$ can be obtained by solving the following optimization objective function:

$$\min_{\mathbf{U},\mathbf{V}} \left\| \mathbf{A} - \mathbf{U}\mathbf{V}\mathbf{U}^T \right\|_F^2,$$

where $\mathbf{U}$ is the low rank matrix and matrix $\mathbf{V}$ saves the correlation among the rows of $\mathbf{U}$, $\|\mathbf{X}\|_F$ is the Frobenius norm of matrix $\mathbf{X}$.

To avoid overfitting, regularization terms $\|\mathbf{U}\|_F^2$ and $\|\mathbf{V}\|_F^2$ are added to the object function as follows [46]:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \left\| \mathbf{A} - \mathbf{U}\mathbf{V}\mathbf{U}^T \right\|_F^2 + \alpha \|\mathbf{U}\|_F^2 + \beta \|\mathbf{V}\|_F^2, \\ &s.t., \mathbf{U} \geq \mathbf{0}, \mathbf{V} \geq \mathbf{0}, \end{aligned}$$

where α and β are the weight of terms $\|\mathbf{U}\|_F^2$, $\|\mathbf{V}\|_F^2$ respectively.

This object function is very hard to achieve the global optimal result for both $\mathbf{U}$ and $\mathbf{V}$. A alternative optimization schema can be used here, which can update $\mathbf{U}$ and $\mathbf{V}$ alternatively. The Lagrangian function of the object equation should be [46]:

$$\begin{aligned}\mathcal{F} &= Tr(\mathbf{A}\mathbf{A}^T) - Tr(\mathbf{A}\mathbf{U}\mathbf{V}^T\mathbf{U}^T) \\ &\quad - Tr(\mathbf{U}\mathbf{V}\mathbf{U}^T\mathbf{A}^T) + Tr(\mathbf{U}\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T\mathbf{U}^T) \\ &\quad + \alpha Tr(\mathbf{U}\mathbf{U}^T) + \beta Tr(\mathbf{V}\mathbf{V}^T) - Tr(\mathbf{\Theta}\mathbf{U}) - Tr(\mathbf{\Omega}\mathbf{V})\end{aligned}$$

where $\mathbf{\Theta}$ and $\mathbf{\Omega}$ are the multiplier for the constraint of $\mathbf{U}$ and $\mathbf{V}$ respectively.

By taking derivatives of $\mathcal{F}$ with regarding to $\mathbf{U}$ and $\mathbf{V}$ respectively, the partial derivatives of $\mathcal{F}$ will be

$$\begin{aligned}\frac{\partial \mathcal{F}}{\partial \mathbf{U}} &= -2\mathbf{A}^T\mathbf{U}\mathbf{V} - 2\mathbf{A}\mathbf{U}\mathbf{V}^T + 2\mathbf{U}\mathbf{V}^T\mathbf{U}^T\mathbf{U}\mathbf{V}^T \\ &\quad + 2\mathbf{U}\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T + 2\alpha\mathbf{U} - \mathbf{\Theta}^T \\ \frac{\partial \mathcal{F}}{\partial \mathbf{V}} &= -2\mathbf{U}^T\mathbf{A}\mathbf{U} + 2\mathbf{U}^T\mathbf{U}\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\beta\mathbf{V} - \mathbf{\Omega}^T\end{aligned}$$

Let $\frac{\partial \mathcal{F}}{\partial \mathbf{U}} = \mathbf{0}$ and $\frac{\partial \mathcal{F}}{\partial \mathbf{V}} = \mathbf{0}$ and use the KKT complementary condition, we can get [46]:

$$\begin{aligned}\mathbf{U}(i,j) &\leftarrow \mathbf{U}(i,j)\sqrt{\frac{\left(\mathbf{A}^T\mathbf{U}\mathbf{V} + \mathbf{A}\mathbf{U}\mathbf{V}^T\right)(i,j)}{\left(\mathbf{U}\mathbf{V}^T\mathbf{U}^T\mathbf{U}\mathbf{V} + \mathbf{U}\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T + \alpha\mathbf{U}\right)(i,j)}}, \\ \mathbf{V}(i,j) &\leftarrow \mathbf{V}(i,j)\sqrt{\frac{\left(\mathbf{U}^T\mathbf{A}\mathbf{U}\right)(i,j)}{\left(\mathbf{U}^T\mathbf{U}\mathbf{V}\mathbf{U}^T\mathbf{U} + \beta\mathbf{V}\right)(i,j)}}.\end{aligned}$$

The low-rank matrix $\mathbf{U}$ captures the information of each users from the adjacency matrix. Each row of $\mathbf{U}$ represents the *latent feature vectors* of users in the network, which can be used in many link prediction models, e.g., supervised link prediction models that will be introduced in the next chapter.

Chapter 4

Link Prediction for Heterogeneous Networks

Many social networks involving abundant information can be represented as heterogeneous networks. In this chapter, we will introduce some classic method and some newly proposed methods for link prediction in heterogeneous networks, which include meta path based *supervised link prediction* methods and collective link prediction framework in heterogeneous networks.

There exists many different heterogeneous social networks in the world but to narrow the domain, we will use location-based social network, Foursquare, as the link prediction target network, e.g., $G = (U \cup L, E_s \cup E_l)$, where U and L are the sets of users and locations, E_s and E_l are sets of the social link and location link sets.

4.1 Supervised Link Prediction

Supervised link prediction models first proposed in [24] has been widely used to solve many link prediction problems [24, 39, 6, 48]. Supervised link prediction models have two important components: feature extraction and classification.

4.1.1 Feature Extraction

In heterogeneous social networks, different kinds of features can be extracted from the network. For example, from existing social connections among users, all the proximity measures among users introduced in previous subsection can be used as features, e.g., (1) features based on local neighbor information, like common neighbor, Jaccard's coefficient; (2) feature based on global path information, like shortest path and Katz; (3) features based on random walk, like commute time and random walk with restart proximity measure; (4) latent feature vectors obtained from matrix factorization. Besides these features extracted

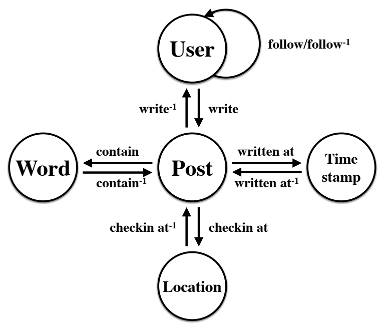


Figure 4.1: Schema of heterogeneous network.

from social connection information, from other types of information, e.g., location check-ins in location based social networks like Foursquare [30, 60, 61], job information in professional social networks like Linkedin [26], tweets information in Twitter [27] and video information in Youtube [14], various heterogeneous features can be extracted for social link prediction tasks. These features can be too diverse that it is nearly impossible to introduce all of them in this survey.

In this part, we will extract a set of generalized features based on *meta path* [44, 45], which can be applied to other heterogeneous networks.

Social Meta Path

Users in heterogeneous online social network can be extensively connected to each other via different paths. In this part, we will categorize the diverse paths connection users within one single network with the *social meta paths* concept [44, 45].

For a given heterogeneous online social network, e.g., G, to describe its structure more clearly, whose *schema* is defined to be $S_G = (T, R)$, where T, R are the sets of node types and link types in G. For example, if $G = (V, E)$, where $V = U \cup L$ contain user and location nodes, $E = E_{u,u} \cup E_{u,l}$ contains the social links and location links, then $S_G = (T, R)$, $T = \{\text{User, Location}\}$ and $R = \{\text{Social Link, Location Link}\}$. A complete schema of the Foursquare network is shown in Figure 4.1. In network G, nodes can be connected with each other via extensive paths consisting of various links. To categorize all possible paths in heterogeneous networks G, the concept of *meta path* based on schema S_G is defined as follows [44, 45]:

Definition 7 (Meta Path): Based on the given the network schema, $S_G = (T, R)$, $\Phi = T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{k-1}} T_k$ is defined to be the *meta path* of network G, where $T_i \in T, i \in \{1, 2, \cdots, k\}$ and $R_i \in R, i \in \{1, 2, \cdots, k-1\}$.

Meanwhile, meta paths can be divided into two different categories depending on types of nodes and links that constitute them.

Definition 8 (Homogeneous and Heterogeneous Meta Path): For a given meta path $\Phi = T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{k-1}} T_k$ defined based on S_G, if $(T_1, \cdots, T_k$ are all the same $) \wedge (R_1, \cdots, R_{k-1}$ are all the same), then Φ is a *homogeneous meta path*; otherwise P is a *heterogeneous meta path*.

In this part, we are mainly concerned about meta paths connecting user nodes, which can be defined as the *social meta path*.

Definition 9 (Social Meta Path): For a given meta path $\Phi = T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{k-1}} T_k$ defined based on S_G, if T_1 and T_k are both the "User", then P is defined as a *social meta path*. Depending on whether $T_1, \cdots, T_k$ and $R_1, \cdots, R_{k-1}$ are the same or not, P can be divided into two categories: *homogeneous social meta path* and *heterogeneous social meta path*.

Based on the schema of the Foursquare networks as shown in Figure 4.1, many different kinds of *homogeneous and heterogeneous social meta paths* for network G can be defined, whose physical meanings and notations are listed as follows:

Homogeneous Social Meta Path

- *ID 0. Follow*: User $\xrightarrow{follow}$ User, whose notation is "$U \to U$" or $\Phi_0(U, U)$.
- *ID 1. Follower of Follower*: User $\xrightarrow{follow}$ User $\xrightarrow{follow}$ User, whose notation is "$U \to U \to U$" or $\Phi_1(U, U)$.
- *ID 2. Common Out Neighbor*: User $\xrightarrow{follow}$ User $\xrightarrow{follow^{-1}}$ User, whose notation is "$U \to U \leftarrow U$" or $\Phi_2(U, U)$.
- *ID 3. Common In Neighbor*: User $\xrightarrow{follow^{-1}}$ User $\xrightarrow{follow}$ User, whose notation is "$U \leftarrow U \to U$" or $\Phi_3(U, U)$.

Heterogeneous Social Meta Path

- *ID 4. Common Words*: User $\xrightarrow{write}$ Post $\xrightarrow{contain}$ Word $\xrightarrow{contain^{-1}}$ Post $\xrightarrow{write^{-1}}$ User, whose notation is "$U \to P \to W \leftarrow P \leftarrow U$" or $\Phi_4(U, U)$.
- *ID 5. Common Timestamps*: User $\xrightarrow{write}$ Post $\xrightarrow{contain}$ Time $\xrightarrow{contain^{-1}}$ Post $\xrightarrow{write^{-1}}$ User, whose notation is "$U \to P \to T \leftarrow P \leftarrow U$" or $\Phi_5(U, U)$.
- *ID 6. Common Location Checkins*: User $\xrightarrow{write}$ Post $\xrightarrow{attach}$ Location $\xrightarrow{attach^{-1}}$ Post $\xrightarrow{write^{-1}}$ User, whose notation is "$U \to P \to L \leftarrow P \leftarrow U$" or $\Phi_6(U, U)$.

Social Meta Path-based Heterogeneous Features

Meta paths introduced in the previous part can actually cover a large number of path instances connecting users in the network. Formally, the fact that node n (or link l) is an instance of node type T (or link type R) in the network can be denoted as $n \in T$ (or $l \in R$). Identity function $I(a, A) = \begin{cases} 1, & \text{if } a \in A \\ 0, & otherwise, \end{cases}$ can check whether node/link a is an instance of node/link type A in the network. The *Social Meta Path based Features* are defined to be:

Definition 10 (Social Meta Path based Features): For a given link (u, v), the feature extracted for it based on meta path $\Phi = T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{k-1}} T_k$ from the network is defined to be the expected number of formed paths between u and v in the network:

$$x(u, v) = I(u, T_1) I(v, T_k) \sum_{n_1 \in \{u\}, n_2 \in T_2, \cdots, n_k \in \{v\}} \prod_{i=1}^{k-1} I((n_i, n_{i+1}), R_i).$$

Features extracted based on $\Phi = \{\Phi_1, \cdots, \Phi_6\}$ are named as the *social meta path* based social features.

4.1.2 Classification Algorithms

Based on meta paths $\{\Phi_1, \cdots, \Phi_6\}$, a set of features for links can be extracted from the network, denoted as $\mathbf{x} = [x_{\Phi_1}, \cdots, x_{\Phi_6}]$. According to the physical meanings, links in social networks can be labeled as positive and negative links, e.g., friends vs. enemies [52], trust vs. distrust [55], positive attitude vs. negative attitude [56] etc. For example, given a directed social link (u, v) in the network, if u distrust v, then $y(u, v) = -1$; otherwise, $y(u, v) = 1$. For the given feature label pairs, different classification algorithms can be used for supervised link prediction. In this part, we will introduce SVM [10] as an example of the classification algorithms [36].

SVM aims at finding the following linear function

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b,$$

where $f : \mathbb{R}^{|\mathbf{x}|} \to \mathbb{R}$ maps a vector to a real value, $\mathbf{w} \in \mathbb{R}^{|\mathbf{x}|}$ is a weight vector and $b \in \mathbb{R}$ is called the bias.

Function f can separate positive instances and negative instances well based f, the label of a link given its feature vector $\mathbf{x}_i$ can be determined according to equation:

$$y_i = \begin{cases} +1, & \text{if } \mathbf{w}^T \mathbf{x}_1 + b \geq 0, \\ -1, & \text{if } \mathbf{w}^T \mathbf{x}_1 + b < 0. \end{cases}$$

In other words, the hyperplane $\mathbf{w}^T \mathbf{x}_1 + b = 0$ is the *decision boundary* desired by SVM. As introduced in [36], given training instances $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\}$ which are linearly separable, the optimal weight vector $\mathbf{w}$ can be obtained by solving the following equation:

$$\min_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{w}}{2},$$
$$s.t., y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 0, i \in \{1, 2, \cdots, n\}.$$

In the real-world situations, the data is usually noisy which can make the SVM proposed above fail to work well as SVM cannot find a solution of the optimization equation because the constraints cannot be satisfied. To solve this problem, a *slack variable*, $\xi_i \geq 0$, can be introduced to relax the strict constraint:

$$
\begin{aligned}
&y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, i \in \{1, 2, \cdots, n\},\\
&\xi_i \geq 0, i \in \{1, 2, \cdots, n\}.
\end{aligned}
$$

To avoid large *slack variables*, penalty term of ξ_i is also added to the target objective function

$$
\begin{aligned}
&\min_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{w}}{2} + c\left(\sum_{i=1}^{n} \xi_i\right)^k,\\
&s.t., y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 0, i \in \{1, 2, \cdots, n\},\\
&\quad \xi_i \geq 0, i \in \{1, 2, \cdots, n\},
\end{aligned}
$$

where $k = 1$ is usually used.

For instances that are not linearly separable, the SVM with linear hyperplane may fail to work, which can be solved by *SVM with nonlinear kernels*. More detailed information about the kernel techniques is introduced in [36].

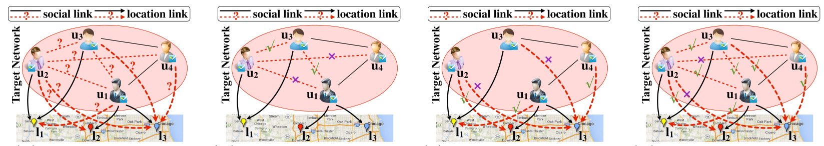
4.2 Collective Link Prediction

Heterogeneous social networks can usually contain multiple kinds of links. As proposed in [61], multiple link prediction tasks in social networks can be strongly correlated and mutually influential to each other. As a result, multiple link prediction tasks in heterogeneous can be done simultaneously. In this part, the collective link prediction problem in heterogeneous networks will be introduced, where both *social links* among users and *location links* between users and locations are to be predicted. The link prediction models used in this part are supervised link prediction models.

Let G be the network studied in this part. The set of users and locations in G are denoted as U and L, while the sets of existing social links and location links in G^t are represented as $\mathcal{E}_s$ and $\mathcal{E}_l$. Collective link prediction problems aim at predicting are a subset of potential social links among users in G^t: $\mathcal{L}_s \subset (U \times U - \mathcal{E}_s)$ and a subset of potential location links in G: $\mathcal{L}_l \subset (U \times L - \mathcal{E}_l)$. In other words,collective link prediction tasks want to build a mapping: $f_M : \{\mathcal{L}_s, \mathcal{L}_l\} \to \{-1, 1\}$ to decide whether potential links in $\{\mathcal{L}_s, \mathcal{L}_l\}$ exist or not and a confidence score function $g_M : \{\mathcal{L}_s, \mathcal{L}_l\} \to [0, 1]$ denoting their existence probabilities.

4.2.1 Correlation Between Different Tasks

When predicting a link, the classifiers will give a score within range $[0, 1]$ to show its existence probability. Newly predicted social links will update the



(a) input network (b) social link prediction (c) location link prediction (d) collective link prediction

Figure 4.2: An example of different link prediction methods. (a) is the input network. (b)-(c) is independent social link and location link prediction result. (d) shows the collective link prediction result.

social link existence probability information in the network, which can affect other location link prediction tasks. For example, these updated social link existence probabilities can change the extended common neighbors of a location and a user. Similarly, the location link prediction task can also influence the social link prediction result.

For example, in Figure 4.2, an example of different link prediction methods is shown. Figure 4.2(a) is the input aligned networks, in which there are 4 users and some existing social links $(u_3, u_4), (u_1, u_4)$ and location links (u_2, l_1), (u_3, l_1), (u_1, l_2), (u_1, l_3) as well as many other potential links to be predicted. Based on the information in the network, including social information (e.g., common neighbors), location information (e.g., co-checkins) and other auxiliary information, traditional link prediction methods can predict social links and locate links independently. Figure 4.2(b) shows the result of independent social link prediction result, in which social link (u_2, u_3) and (u_1, u_3) are predicted to be existent, while social link (u_1, u_2) and (u_2, u_4) are predicted to be nonexistent. Figure 4.2(c) shows the independent location link prediction result and in the result, location links $(u_2, l_2), (u_1, l_1), (u_4, l_3)$ are predicted to be existent, while (u_2, l_3) and (u_3, l_3) is predicted to be nonexistent.

From the results in Figures 4.2(b) and 4.2(c), some problematic phenomena can be found. For example, user u_2 and u_1 are predicted to have visited locations l_1, l_2 and they are also predicted to share a common neighbor: u_3. Based on the result, it is highly likely that the potential social link (u_2, u_3) will be predicted to be existent. However, it is predicted to be nonexistent in Figure 4.2(b). Another example is that many neighbors of user u_3, both the originally existing u_4 and the newly predicted u_1 both have visited or are predicted to have visited l_3. By using Friend-based Collaborative Filtering (FCF) [57], u_3 is highly likely to be predicted to have visited l_3. However, the location link between u_3 and l_3 is predicted to be nonexistent in Figure 4.2(c).

With consideration of the correlation between these two link prediction tasks and predict social links and location links simultaneously, the predicted results of social link (u_1, u_2) and location link (u_3, l_3) are highly likely to be predicted as existent. In Figure 4.2(d), a potential result of collective link prediction methods is shown.

4.2.2 Collective Link Prediction

We formulate the sets of potential social links and potential location links to be predicted as $\mathcal{L}_s$ and $\mathcal{L}_l$ in the problem formulation section. For links $l_s^t \in \mathcal{L}_s$ and $l_l^t \in \mathcal{L}_l$, the supervised models built with the existing information in the network will give them the predicted labels: $y(l_s^t)$ and $y(l_l^t)$, as well as the existence probability scores: $P(y(l_s^t) = 1)$ and $P(y(l_l^t) = 1)$. Traditional methods predicting social links and location links independently aims at finding the set of labels achieving the maximum probability scores for each kind of links. In other words, let $\hat{\mathcal{Y}}_s \subset \{-1, 1\}^{|\mathcal{L}_s|}$, $\hat{\mathcal{Y}}_l \subset \{-1, 1\}^{|\mathcal{L}_l|}$ be the sets of optimal labels

$$\hat{\mathcal{Y}}_s = arg \max_{\mathcal{Y}_s} P(\mathbf{y}(\mathcal{L}_s) = \mathcal{Y}_s)$$

$$\hat{\mathcal{Y}}_l = arg \max_{\mathcal{Y}_l} P(\mathbf{y}(\mathcal{L}_l) = \mathcal{Y}_l)$$

where, $P(\mathbf{y}\mathcal{L}_s) = \mathcal{Y}_s)$ and $P(\mathbf{y}(\mathcal{L}_l) = \mathcal{Y}_l)$ denote the probability scores achieved when links in $\mathcal{L}_s$ and $\mathcal{L}_l$ are assigned with labels in $\mathcal{Y}_s$ and $\mathcal{Y}_l$.

However, considering connections between these two link prediction tasks, the inferred social link or location link information should all be used in other link prediction tasks. The optimal selection of label sets $\hat{\mathcal{Y}}_s$ and $\hat{\mathcal{Y}}_l$ will be

$$\begin{aligned}\hat{\mathcal{Y}}_s, \hat{\mathcal{Y}}_l = arg \max_{\mathcal{Y}_s, \mathcal{Y}_l} & P(\mathbf{y}(\mathcal{L}_s) = \mathcal{Y}_s | \mathbf{y}(\mathcal{L}_l) = \mathcal{Y}_l) \\ & \times P(\mathbf{y}(\mathcal{L}_l) = \mathcal{Y}_l | \mathbf{y}(\mathcal{L}_s) = \mathcal{Y}_s)\end{aligned}$$

For the given optimization equation, there are many different solutions. In this part, an iterative method, TRAIL, is proposed [61] to approach it, which can predict the social links and location links iteratively until convergence. Let τ be the τ_{th} iteration and the optimal label sets of social links and location links achieved in the τ_{th} iteration be $\hat{\mathcal{Y}}_s^{(\tau)}$ and $\hat{\mathcal{Y}}_l^{(\tau)}$, then

$$\begin{aligned}\hat{\mathcal{Y}}_s^{(\tau)} &= arg \max_{\mathcal{Y}_s} P(\mathbf{y}(\mathcal{L}_s) = \mathcal{Y}_s | G^t, \mathbf{y}(\mathcal{L}_s) = \hat{\mathcal{Y}}_s^{(\tau-1)}, \\ &\qquad \mathbf{y}(\mathcal{L}_l) = \hat{\mathcal{Y}}_l^{(\tau-1)}) \\ \hat{\mathcal{Y}}_l^{(\tau)} &= arg \max_{\mathcal{Y}_l} P(\mathbf{y}(\mathcal{L}_l) = \mathcal{Y}_s | G^t, \mathbf{y}(\mathcal{L}_s) = \hat{\mathcal{Y}}_s^{(\tau)}, \\ &\qquad \mathbf{y}(\mathcal{L}_l) = \hat{\mathcal{Y}}_l^{(\tau-1)})\end{aligned}$$

Chapter 5

Link Prediction across Aligned Networks

Nowadays, users are usually involved in multiple aligned social networks at the same time. Link prediction with multiple sources first proposed in [39] has become a hot research topic in recent years. Meanwhile, these networks sharing common users are formulated as aligned heterogeneous networks in [30, 60, 61].

Given two aligned heterogeneous networks G^i and G^j, if all user accounts in one network are related to accounts in the other network by anchor links mutually, then G^i and G^j are *fully aligned*, in which case $|U^i| = |U^j| = |A^{i,j}|$ and the anchor links in $A^{i,j}$ have an inherent *one-to-one* property [4]. While, if some users in G^i do not have the corresponding accounts in G^j or some users in G^j do not have the corresponding accounts in G^i, then G^i and G^j are *partially aligned* and $|A^{i,j}| \leq \min\{|U^i|, |U^j|\}$.

Link prediction across multiple aligned heterogeneous networks has just been proposed by Zhang et al. [30, 60, 61] in recent years. In this part, we will introduce anchor link prediction at first. Then, we will introduce a link prediction method with strict co-existence information transfer across networks [60, 61], which can transfer useful information for anchor users from aligned networks.

5.1 Anchor Link Prediction

Suppose we have two heterogeneous social networks G^s and G^t, with a small set of known anchor links between the users accounts in two networks, $A = \{(u_i^s, u_j^t), u_i^s \in U^s, u_j^t \in U^t\}$. Anchor links are one-to-one relationships between user accounts in U^s and U^t, *i.e.*, no two anchor links share a same user account. (u_i^s, u_j^t) denotes that the two user accounts belong to the same user. The task of anchor link prediction is to predict whether there is an anchor link between a pair of user accounts u_i^s and u_j^t, where $u_i^s \in U^s, u_j^t \in U^t$. challenges that make our problem

The key issue of *anchor link prediction* is to learn a one-to-one matching between the user accounts of two heterogeneous social networks. This problem formulation is different from existing works on social link prediction [21, 34, 24, 49, 35] mainly in two-folds: First, the target links to predict are one-to-one relationships between two sets of nodes, *e.g.*, Twitter accounts and Facebook accounts. How can we extract informative features for anchor link prediction task? Existing features for link prediction, such as number of common neighbors and the shortest distance, require that the target links should be many-to-many relationships. Second, the prediction of all anchor links should be considered collectively due to the one-to-one constraint. Supervised link prediction methods usually make predictions on a set of links independently, because there is no constraint on the degree of each node in the network.

A two-phase link prediction method is proposed in [30], where the first phase tackles feature extraction problem, while the second phase takes care of one-to-one constrained anchor link prediction. Next, these two phases will be introduced one by one.

5.1.1 Heterogeneous Feature Extraction across Networks

As proposed in [30], from heterogeneous networks, different kinds of features can be extracted, which include the extended definitions of "*common neighbors*", "*Jaccard's coefficient*" and "*Adamic/Adar measure*" [2].

• **Extended Common Neighbors**: $CN(u_i^s, u_j^t)$ represents the number of 'common' neighbors between u_i^s in the source network and u_j^t in the target network. The neighbors of u_i^s in the source network can be denoted as $\Gamma_s(u_i^s)$ and the neighbors of u_j^t in the target network can be denoted as $\Gamma_t(u_i^t)$. The measure of *extended common neighbor* is defined as the number of known anchor links between $\Gamma_s(u_i^s)$ and $\Gamma_t(u_i^t)$.

$$\begin{aligned} CN(u_i^s, u_j^t) &= \left|\{(u_p^s, u_q^s) \in A, u_p^s \in \Gamma_s(u_i^s), u_q^s \in \Gamma_t(u_i^t)\}\right| \\ &= \left|\Gamma_s(u_i^s) \bigcap_A \Gamma_t(u_i^t)\right| \end{aligned}$$

It indicates how many pairs of user accounts belong to a same user.

• **Extended Jaccard's coefficient**: The extended measure of Jaccard's coefficient to multi-network setting is defined using similar method of extending common neighbor. $JC(u_i^s, u_j^t)$ is a normalized version of common neighbors, *i.e.*, $CN(u_i^s, u_j^t)$ divided by the total number of distinct users in $\Gamma_s(u_i^s) \cup \Gamma_t(u_j^t)$:

$$JC(u_i^s, u_j^t) = \frac{|\Gamma_s(u_i^s) \bigcap_A \Gamma_t(u_i^t)|}{|\Gamma_s(u_i^s) \bigcup_A \Gamma_t(u_i^t)|}$$

where

$$\left|\Gamma_s(u_i^s) \bigcup_A \Gamma_t(u_i^t)\right| = |\Gamma_s(u_i^s)| + |\Gamma_t(u_i^t)| - \left|\Gamma_s(u_i^s) \bigcap_A \Gamma_t(u_i^t)\right|$$

• **Extended Adamic/Adar Measure**: Similarly, the extended the Adamic/Adar Measure is defined into multi-network settings, where the common neighbors are weighted by their average degrees in both social networks.

$$AA(u_i^s, u_j^t) = \sum_{\forall (u_p^s, u_q^s) \in \Gamma_s(u_i^s) \bigcap_A \Gamma_t(u_i^t)} \log^{-1}\left(\frac{|\Gamma_s(u_p^s)| + |\Gamma_t(u_q^t)|}{2}\right).$$

5.1.2 Inferring Anchor Links w.r.t. One-to-one Constraint

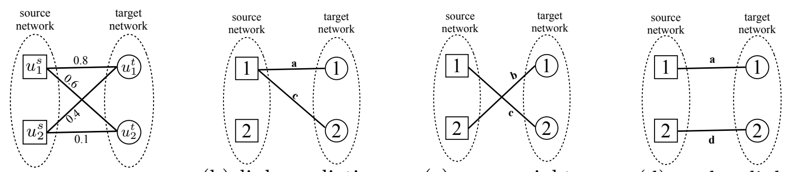
After extracting all the four types of heterogeneous features in the previous section, a binary classifier can be trained, such as SVM or logistic regression, for anchor link prediction. However, in the inference process, the predictions of the binary classifier cannot be directly used as anchor links due to the following issues:

- The inference of conventional classifiers are designed for constraint-free settings, and the one-to-one constraint may not necessarily hold in the label prediction of the classifier (SVM).
- Most classifiers also produce output scores, which can be used to rank the data points in the test set. However, these ranking scores are uncalibrated in scale to anchor link prediction task. Previous classifier calibration methods [59] apply only to classification problems without any constraint.

In order to tackle the above issues, a novel inference process, called PUclf (Multi-Network Anchoring), to infer anchor links based upon the ranking scores of the classifier is introduced in [30], which is motivated by the *stable marriage problem* [16] in mathematics.

Before introduce the method, a toy example is shown in Figure 5.1 to illustrate the main idea of our solution. Suppose in Figure 5.1(a) the ranking scores from the classifiers are given. As shown in Figure 5.1(b), link prediction methods with a fixed threshold may not be able to predict well, because the predicted links do not satisfy the constraint of one-to-one relationship. Thus one user account in the source network can be linked with multiple accounts in the target network. In Figure 5.1(c), *weighted maximum matching* methods can find a set of links with maximum sum of weights. However, it is worth noting that the input scores are uncalibrated, so maximum weight matching may not be a good solution for anchor link prediction problems. The input scores only indicate the ranking of different user pairs, *i.e.*, the preference relationship among different user pairs.

Here we say 'node x prefers node y over node z', if the score of pair (x, y) is larger than the score of pair (x, z). For example, in Figure 5.1(c), the weight of pair a, *i.e.*, Score$(a) = 0.8$, is larger than Score$(c) = 0.6$. It shows that user u_1^s (the first user in the source network) *prefers* u_1^t over u_2^t. The problem with the prediction result in Figure 5.1(c) is that, the pair (u_1^s, u_1^t) should be more likely to be an anchor link due to the following reasons: (1) u_1^s prefers u_1^t over u_2^t; (2) u_1^t also prefers u_1^s over u_2^s.



(a) input scores (b) link prediction (c) max weights (d) anchor link

Figure 5.1: An example of anchor link inference by different methods. (a) is the input, ranking scores. (b)-(d) are the results of different methods for anchor link inference.

Definition 11 (Blocking Pair): A pair (u_i^s, u_j^t) is a blocking pair iff u_i^s and u_j^t both prefer each other over their current assignments respectively in the predicted set of anchor links A'.
Definition 12 (Stable Matching): An inferred anchor link set A' is stable if there is no blocking pair.

The anchor link prediction problem is formulated as a stable matching problem between user accounts in source network and accounts in target network [30]. Assume that we have two sets of unlabeled user accounts, *i.e.*, $U^s = \{u_i^s\}_i$ in source network and $U^t = \{u_j^t\}_j$ in target network. Each u_i^s has a ranking list or preference list $P(u_i^s)$ over all the user accounts in target network ($u_j^t \in U^t$) based upon the input scores of different pairs. For example, in Figure 5.1(a), the preference list of node u_1^s is $P(u_1^s) = (u_1^t > u_2^t)$, indicating that node u_1^t is preferred by u_1^s over u_2^t. The preference list of node u_2^s is also $P(u_2^s) = (u_1^t > u_2^t)$. Similarly, a preference list can be built for each user account in the target network. In Figure 5.1(a), $P(u_1^t) = P(u_2^t) = (u_1^s > u_2^s)$.

The proposed PUclf method for anchor link prediction is shown in Algorithm 1. In each iteration, a free user account u_i^s is randomly selected from the source network. Then the most preferred user node u_j^t by u_i^s in its preference list $P(u_i^s)$ is obtained. u_j^t is then removed from the preference list, *i.e.*, $P(u_i^s) = P(u_i^s) - u_j^t$.

If u_j^t is also a free account, the pair of accounts (u_i^s, u_j^t) is added into the current solution set A'. Otherwise, u_j^t is already occupied with u_p^s in A'. We then examine the preference of u_j^t. If u_j^t also prefers u_i^s over u_p^s, it means that the pair (u_i^s, u_j^t) is a blocking pair. The blocking pair is removed by replacing the pair (u_p^s, u_j^t) in the solution set A' with the pair (u_i^s, u_j^t). Otherwise, if u_j^t prefers u_p^s over u_i^s, the the next iteration is started to reach out the next free node in the source network. The algorithm stops when all the users in the source network are occupied, or all the preference lists of free accounts in the source network are empty.

Algorithm 1 Multi-Network Anchoring

Input: two heterogeneous social networks, G^s and G^t.
a set of known anchor links A

Output: a set of inferred anchor links A'

Construct a training set of user account pairs with known labels using A.
For each pair (u_i^s, u_j^t), extract four types of features.
Training classification model C on the training set.
Perform classification using model C on the test set.
For each unlabeled user account, sort the ranking scores into a preference list of the matching accounts.
Initialize all unlabeled u_i^s in G^s and u_j^t in G^t as free
$A' = \emptyset$
while $\exists$ free u_i^s in G^s and u_i^s's preference list is non-empty **do**
  Remove the top-ranked account u_j^t from u_i^s's preference list
  if u_j^t is free **then**
    $A' = A' \cup \{(u_i^s, u_j^t)\}$
    Set u_i^s and u_j^t as occupied
  else
    $\exists u_p^s$ that u_j^t is occupied with.
    if u_j^t prefers u_i^s to u_p^s **then**
      $A' = (A' - \{(u_p^s, u_j^t)\}) \cup \{(u_i^s, u_j^t)\}$
      Set u_p^s as free and u_i^s as occupied
    end if
  end if
end while

5.2 Link Transfer across Aligned Networks

5.2.1 Supervised Link Prediction

Traditional supervised link prediction methods by using one single network implicitly or explicitly assume that information in the target network itself is enough to build effective link prediction models. These methods use the extracted features of existing links in the target network to train classifiers, which will be applied to predict other potential links. For example, the existence probability of a social link (u_i^t, u_j^t) in the target network G^t can be predicted to be:

$$P(y(u_i^t, u_j^t) = 1 | G^t)$$

where $y(u_i^t, u_j^t)$ is the label of link (u_i^t, u_j^t). From G^t, a set of heterogeneous features can be extracted for social link (u_i^t, u_j^t). Then

$$P(y(u_i^t, u_j^t) = 1 | G^t) = P(y(u_i^t, u_j^t) = 1 | \boldsymbol{x}(u_i^t, u_j^t))$$

where $\boldsymbol{x}(u_i^t, u_j^t) = [x(u_i^t, u_j^t)^1, x(u_i^t, u_j^t)^2, \cdots, x(u_i^t, u_j^t)^n]^T, n = |\boldsymbol{x}(u_i^t, u_j^t)|$ and $x(u_i^t, u_j^t)^k, k \in \{1, 2, \cdots, n\}$ is the k_{th} feature extracted from the target network for social link (u_i^t, u_j^t). Usually, feature $x(u_i^t, u_j^t)^k$ can be the summarized properties of social link (u_i^t, u_j^t), e.g., extended common neighbors.

Similarly, for a certain location link (u_i^t, l_j^t) in G^t, the extracted features can be used for it from the target network, $\boldsymbol{x}(u_i^t, l_j^t)$, to predict its existence probability.

$$P(y(u_i^t, l_j^t) = 1|G^t) = P(y(u_i^t, l_j^t) = 1|\boldsymbol{x}(u_i^t, l_j^t))$$

If the target network is quite new, the features vectors extracted for both social links and location links can be very sparse, which can hardly build good link prediction models. Next, information transferred from the aligned source network can be used to solve the problem.

5.2.2 Link Transfer across Aligned Networks

With the *anchor links*, users' corresponding accounts in the aligned source network can be located, information in which can be transferred to the target network. Suppose, for instance, we want to predict a potential social link (u_i^t, u_j^t) by using information in both networks. By taking advantages of the *anchor links*, the corresponding accounts of u_i^t and u_j^t in the aligned source network can be obtained: u_i^s and u_j^s. If u_i^s and u_j^s both exist in G^s, then information related to the corresponding social link (u_i^s, u_j^s) in the aligned source network can be transferred to the target network, which is represented as a feature vector extracted from G^s for link (u_i^s, u_j^s): $\boldsymbol{x}(u_i^s, u_j^s)$. Noticing that the existence information of link (u_i^s, u_j^s) in the aligned source network, $y(u_i^s, u_j^s)$, can be very useful, it is defined as *pseudo label* of link (u_i^t, u_j^t).

Definition 13 (Pseudo Label): Let (n_i^t, n_j^t) be a link in G^t, where n_i^t, n_j^t are nodes in it and they can be users, locations, etc., the corresponding link of (n_i^t, n_j^t) in the aligned source network G^s will be (n_i^s, n_j^s). The existence indicator of link (n_i^s, n_j^s) in G^s: $y(n_i^s, n_j^s)$ is defined as the *pseudo label* of link (n_i^t, n_j^t).

The *pseudo label* is used as an extra feature added to the extended feature vector, obtained by merging feature vectors extracted from G^t and G^s.

$$\begin{aligned} &P(y(u_i^t, u_j^t) = 1|G^t, G^s) \\ &= P\left(y(u_i^t, u_j^t) = 1| \left[\boldsymbol{x}(u_i^t, u_j^t)^T, \boldsymbol{x}(u_i^s, u_j^s)^T, y(u_i^s, u_j^s)\right]^T\right) \end{aligned}$$

Similarly, for a certain location link (u_i^t, l_j^t), we have

$$\begin{aligned} &P(y(u_i^t, l_j^t) = 1|G^t, G^s) \\ &= P\left(y(u_i^t, l_j^t) = 1| \left[\boldsymbol{x}(u_i^t, l_j^t)^T, \boldsymbol{x}(u_i^s, l_j^s)^T, y(u_i^s, l_j^s)\right]^T\right) \end{aligned}$$

Actually, the *pseudo label* can also be used as the prediction result of link (n_i^t, n_j^t) in G^t and the method is called the Naive, which will be used as a baseline in our experiment.

Chapter 6

Future Works

6.1 Class Imbalance Problem

Supervised link prediction methods introduced in this article may suffer from the class imbalance problem a lot, as the number of unconnected links is almost the square of the number of existing links. This problem can be solved with existing works, e.g., down sampling method [11], cost sensitive techniques [28]. Another promising method to deal with such problem is to apply PU learning techniques [37] to link prediction tasks, where existing and unconnected links are regarded as positive and unlabeled links respectively.

6.2 Information Transfer for Non-anchor Users

Existing link prediction methods [60, 61] can only transfer useful information for anchor users. Transferring useful information for both anchor and non-anchor users can be what we desire. One possible approach to achieve such goal can be information propagation method, e.g., random walk [22, 19, 31, 6, 47], or inter-network meta paths [58, 44]. Another possible method is to extend and adapt traditional transfer learning techniques [41] to the setting of multiple aligned networks.

6.3 Network Difference Problem

Different Networks can have different characteristics and, as a result, information transferred from other networks can be useful for the target network but can be misleading as well, which is called the domain/network difference problem [41, 56]. Selecting useful information, e.g., feature selection [23], or controlling the proportion of misleading information transferred from the aligned networks, e.g., adjusting the weights of different features [51], can be possible methods to solve the negative transfer problem [20].

Bibliography

[1] K. Aditya A. Menon and C. Elkan. Link prediction via matrix factorization. In *ECML/PKDD*, 2011.

[2] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 2001.

[3] A. Menon K. Aditya and C. Elkan. Link prediction via matrix factorization. In *ECML/PKDD*, 2011.

[4] A. Aladag and C. Erten. Spinal: scalable protein interaction network alignment. *Bioinformatics*, 2013.

[5] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *WWW*, 2007.

[6] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, 2011.

[7] A.-L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaboration. In *Physica A*, 2002.

[8] M. Bilgic, G. Namata, and L. Getoor. Combining collective classification and link prediction. In *ICDMW*, 2007.

[9] C. Bliss, M. Frank, C. Danforth, and P. Dodds. An evolutionary algorithm approach to link prediction in dynamic social networks. *CoRR*, 2013.

[10] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[11] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Artificial Intelligence Review*, 2002.

[12] E. Cho, S. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, 2011.

[13] A. Clauset, C. Moore, and M. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191), 2008.

[14] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The youtube video recommendation system. In *RecSys*, 2010.

[15] P. Domingos and M. Richardson. Markov logic: A unifying framework for statistical relational learning. In *ICML Workshop*, 2004.

[16] L. Dubins and D. Freedman. Machiavelli and the gale-shapley algorithm. *The American Mathematical Monthly*, 1981.

[17] D. Dunlavy, T. Kolda, and E. Acar. Temporal link prediction using matrix and tensor factorizations. *TKDD*, 2011.

[18] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, and Y. Elovici. Link prediction in social networks using computationally efficient topological features. In *SocialCom/PASSAT*, 2011.

[19] F. Fouss, A. Pirotte, J. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *TKDE*, 2007.

[20] L. Ge, J. Gao, H. Ngo, K. Li, and A. Zhang. On handling negative transfer and imbalanced distributions in multiple source transfer learning. In *SDM*, 2013.

[21] L. Getoor and C. P. Diehl. Link mining: A survey. *SIGKDD Explorations Newsletter*, 7(2), 2005.

[22] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *HYPERTEXT*, 1998.

[23] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Machine Learning Research*, 2003.

[24] M. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *SDM*, 2006.

[25] M. Hasan and M. Zaki. In *Social Network Data Analytics*. 2011.

[26] C. Hsieh, M. Tiwari, D. Agarwal, X. Huang, and S. Shah. Organizational overlap on social networks and its applications. In *WWW*, 2013.

[27] Y. Jia, Y. Wang, J. Li, K. Feng, X. Cheng, and J. Li. Structural-interaction link prediction in microblogs. In *WWW*, 2013.

[28] G. Karakoulas and J. Shawe-Taylor. Optimizing classifiers for imbalanced training sets. In *NIPS*, 1999.

[29] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 1953.

[30] X. Kong, J. Zhang, and P. Yu. Inferring anchor links across multiple heterogeneous social networks. In *CIKM*, 2013.

[31] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR*, 2009.

[32] A. Korolova, R. Motwani, S. Nabar, and Y. Xu. Link privacy in social networks. In *CIKM*, 2008.

[33] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM*, pages 556–559, 2003.

[34] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM*, 2003.

[35] R. Lichtenwalter, J. Lussier, and N. Chawla. New perspectives and methods in link prediction. In *KDD*, 2010.

[36] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. 2006.

[37] B. Liu, Y. Dai, X. Li, W. Lee, and P. Yu. Building text classifiers using positive and unlabeled examples. In *ICDM*, 2003.

[38] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 2011.

[39] Z. Lu, B. Savas, W. Tang, and I. Dhillon. Supervised link prediction using multiple sources. In *ICDM*, 2010.

[40] G. Namata, S. Kok, and L. Getoor. Collective graph identification. In *KDD*, 2011.

[41] S. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 2010.

[42] S. Purnamrita, C. Deepayan, and J. Michael. Nonparametric link prediction in dynamic networks. In *ICML*, 2012.

[43] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *KDD*, 2011.

[44] Y. Sun, R. Barber, M. Gupta, C. Aggarwal, and J. Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *ASONAM*, 2011.

[45] Y. Sun, J. Han, C. Aggarwal, and N. Chawla. When will it happen?: relationship prediction in heterogeneous information networks. In *WSDM*, 2012.

[46] J. Tang, H. Gao, X. Hu, and H. Liu. Exploiting homophily effect for trust prediction. In *WSDM*, 2013.

[47] H. Tong, C. Faloutsos, and J. Pan. Fast random walk with restart and its applications. In *ICDM*, 2006.

[48] J. Vert and Y. Yamanishi. Supervised graph inference. In *NIPS*, 2005.

[49] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *ICDM*, 2007.

[50] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A. Barabasi. Human mobility, social ties, and link prediction. In *KDD*, 2011.

[51] D. Wettschereck, D. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for aclass of lazy learning algorithms. *Artificial Intelligence Review*, 1997.

[52] K. Wilcox and A. T. Stephen. Are close friends the enemy? online social networks, self-esteem, and self-control. Journal of Consumer Research, 2012.

[53] W. Xi, B. Zhang, Z. Chen, Y. Lu, S. Yan, W. Ma, and E. Fox. Link fusion: a unified link analysis framework for multi-type interrelated data objects. In *WWW*, 2004.

[54] Y. Yang, N. Chawla, Y. Sun, and J. Han. Link prediction in heterogeneous networks: Influence and time matters. In *ICDM*, 2012.

[55] Y. Yao, H. Tong, X. Yan, F. Xu, and J. Lu. Matri: a multi-aspect and transitive trust inference model. In *WWW*, 2013.

[56] J. Ye, H. Cheng, Z. Zhu, and M. Chen. Predicting positive and negative links in signed social networks by transfer learning. In *WWW*, 2013.

[57] M. Ye, P. Yin, and W. Lee. Location recommendation for location-based social networks. In *GIS*, 2010.

[58] X. Yu, Q. Gu, M. Zhou, and J. Han. Citation prediction in heterogeneous bibliographic networks. In *SDM*, 2012.

[59] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, 2002.

[60] J. Zhang, X. Kong, and P. Yu. Predicting social links for new users across aligned heterogeneous social networks. In *ICDM*, 2013.

[61] J. Zhang, X. Kong, and P. Yu. Transferring heterogeneous links across location-based social networks. In *WSDM*, 2014.

[62] T. Zhou, L. Lü, and Y. Zhang. Predicting missing links via local information. *The European Physical Journal B*, 2009.