Link Prediction

7

7.1 Overview

Given a screenshot of the online social networks, the problem of inferring the missing links or the links to be formed in the networks in the future is called the *link prediction* problem. Link prediction problem has concrete applications in the real world, and many concrete services can be cast to the link prediction problem. For instance, the friend recommendations problem [15] in online social networks can be modeled as the friendship link prediction problem among users. Users' trajectory prediction problem [4] can be formulated as the prediction task of potential check-in links between users and offline POIs (points of interest). The user identifier resolution problem [19, 26] across networks (i.e., the network alignment problem introduced in the previous chapters) can be modeled as the anchor link prediction problem of user accounts across different online social networks.

In this chapter, we will take the friendship link as an example to introduce the general link prediction problem in online social networks. Formally, given the training set $\mathcal{T}_{train}$ involving links belong to different classes ($\mathcal{Y} = \{+1, -1\}$, where $+1$ denotes the positive class and -1 denotes the negative class; sometimes we also use 0 to denote the negative class) and the test set $\mathcal{T}_{test}$ (with unknown labels for the links), the link prediction problem aims at building a mapping $f : \mathcal{T}_{train} \cup \mathcal{T}_{test} \rightarrow \mathcal{Y}$ to project these links to their potential labels in $\mathcal{Y}$.

Depending on the scenarios where we study the link prediction problem, the existing link prediction works can be divided into several different categories. Traditional link prediction problems are mainly focused on inferring the links in one single homogeneous network, like inferring the friendship links [62] among users in online social networks or co-author links [39] in bibliographic networks. As the network structures are becoming more and more complicated, many complex network structures can be modeled as the heterogeneous networks involving different types of nodes and complex connections among them. The heterogeneity of the networks leads to many new link prediction problems, like predicting the links between nodes belonging to different categories [58] and the concurrent inference of multiple types of links in the heterogeneous networks [54, 58]. In recent years, many online social networks have appeared, and lots of new research opportunities exist for researchers and practitioners to study the link prediction problems from the cross-network perspective [57, 58, 61].

Meanwhile, depending on the learning settings used in the link prediction problem formulation and models, the existing link prediction works can be categorized into different groups according to the supervision information involved in the model building. For some of the link prediction models, they

J. Zhang, P. S. Yu, *Broad Learning Through Fusions*,
https://doi.org/10.1007/978-3-030-12528-8_7

calculate the user-pair closeness [53] as the link prediction result without any training data, which are referred to as the *unsupervised link prediction models*. For some other models, they will assign the links with different labels, and use them as the training set to learn a supervised classification models as the base model instead. These models are called the *supervised link prediction models* [14]. Usually, manual labeling of the links is very expensive and tedious. In recent years, many of the works have proposed to apply *semi-supervised learning* techniques [54, 59, 61] in the link prediction problem to utilize the links without labels.

In this chapter, we will introduce the social link prediction problems in online social networks. In Sect. 7.2, we will introduce the social link prediction works in one single homogeneous social network, and the models involve the unsupervised models [23], classification based supervised model [14, 25], and the matrix factorization based link prediction model [1]. The link prediction works in the heterogeneous networks will be introduced in Sect. 7.3, including both the supervised link prediction model [14, 25] and the prediction task of multiple types of links [54, 58]. In the following sections, we will talk about the link prediction problem across multiple heterogeneous social networks. In Sect. 7.4, we introduce a novel cross-network social link prediction model to predict social links for new users specifically [57]. In Sect. 7.5, we will focus on introducing the social link prediction model with positive and unlabeled (PU) learning models [54, 59]. Finally, to overcome the domain difference problem, in Sect. 7.6, we will introduce a matrix estimation based social link prediction model with both positive and unlabeled links [61].

7.2 Traditional Single Homogeneous Network Link Prediction

Traditional link prediction problems are mainly studied based on one single homogeneous network, involving one single type of nodes and links. In this section, we will first briefly introduce how to use the social closeness measures [23, 53] introduced in Sect. 3.3.3 for the link prediction tasks. To integrate different social closeness measures together for the link prediction, we will introduce the supervised link prediction model [14]. Some models formulate the link prediction task as a recommendation problem, and propose to apply the matrix factorization method [1] to address the problem. In this section, we will introduce these three types of link prediction models for the traditional one single homogeneous network.

7.2.1 Unsupervised Link Prediction

Given a screenshot of a homogeneous network $G = (\mathcal{V}, \mathcal{E})$, the unsupervised link prediction models [23, 53] introduced here aim at inferring the potential links that will be formed in the future. Usually, the unsupervised link prediction models will calculate some metrics for the links, which will be used as the predicted confidence scores for these links. Depending on the specific scenario and the link formation assumptions applied, different metrics have been proposed for the link prediction tasks already.

Many of the link prediction metrics are based on the assumption that "*close users are more likely to be friends*," and use the social closeness measure as the link prediction confidence score. The social closeness measures introduced in Sect. 3.3.3, like the *local closeness measures* (e.g., common neighbor, Jaccard's coefficient, Adamic/Adar), the *global path based closeness measures* (e.g., shortest path, Katz), and the *random walk based closeness measures* (e.g., hitting time, commute time, and cosine similarity), can all be used to infer potential connections among users.

In this part, we will not talk about these measures again, and the readers may refer to the previous section for more information. Next, we will introduce the general learning settings and evaluation metrics for the link prediction problem with the unsupervised learning models.

7.2.1.1 Unsupervised Link Prediction Problem Setting

Suppose in the network $G = (\mathcal{V}, \mathcal{E})$, each link is associated with a timestamp. For instance, for link $e \in \mathcal{E}$, we can denote the formation timestamp of link e as $t(e)$. Given three time points $t_p < t_c < t_f$ denoting a past time point t_p, the current time point t_c, and a future time point t_f, we can retrieve the network structure formed in the time range $[t_p, t_c]$ as the current network G_{t_p,t_c} and the network structure to be formed in the future time range $(t_c, t_f]$ as the future network G_{t_c,t_f} respectively. The current network can serve as the input of a link prediction algorithm, i.e., G_{t_p,t_c}, which can infer the new connections to be formed in the future time range $(t_c, t_f]$, i.e., the future network structure G_{t_c,t_f}.

As the network structure evolves, new links will be formed and new nodes will be formed as well. However, for a new node to join in the network in the time range $(t_c, t_f]$, we have no historical knowledge about it and can hardly predict links incident to it, which is also referred to as the *cold start* problem [57,63]. Generally, in the link prediction problems, we have a subset of the nodes as the *core set* $\mathcal{V}^c \subset \mathcal{V}$, and we will be focused on studying the links incident to nodes in the *core set* only. For all the new links to be formed in time range $(t_c, t_f]$, we will sample a subset of the links incident to these nodes in the *core set* only to study the link prediction problem. For the cold start link prediction problem regarding the new users, we will address it in Sect. 7.4 specifically.

Given the current network structure G_{t_p,t_c} and the *core set* $\mathcal{V}^c$, we can represent the formed links among the *core set* users as set $\mathcal{E}^c_{t_p,t_c} = \mathcal{E}_{t_p,t_c} \cap \mathcal{V}^c \times \mathcal{V}^c$. Meanwhile, the remaining links among the *core set* users can be represented as set $\mathcal{E}^r_{t_p,t_c} = \mathcal{V}^c \times \mathcal{V}^c \setminus (\{(u,u)\}_{u \in \mathcal{V}^c} \cup \mathcal{E}^c_{t_p,t_c})$. The link prediction model aims inferring: "among all the links in $\mathcal{E}^r_{t_p,t_c}$, which will be formed in the time range $(t_c, t_f]$ and appear in the future network structure G_{t_c,t_f} (i.e., in set $\mathcal{E}^c_{t_c,t_f} = \mathcal{E}_{t_c,t_f} \cap \mathcal{V}^c \times \mathcal{V}^c$)".

7.2.1.2 Unsupervised Link Prediction Models

In the unsupervised link prediction model, we use the *social closeness* as the prediction confidence measure based on the assumption that "*close users tend to be friends*". In Table 7.1, we summarize some closeness measures we have introduced before. For instance, if we use "*Common Neighbor*" as the social closeness measure, we can represent the closeness score of all the user pairs in set $\mathcal{E}^r_{t_p,t_c}$ as $\{C(e)\}_{e \in \mathcal{E}^r_{t_p,t_c}}$.

The scores of all these remaining links, i.e., $\{C(e)\}_{e \in \mathcal{E}^r_{t_p,t_c}}$, can be outputted as the result. Meanwhile, when determining the links to be recommended for each user in the *core set*, we can pick either the *top-k* links with the highest predicted scores or set a threshold to select the links with scores greater than the threshold as the ones to be formed. In other words, these selected links will be assigned with positive labels, while the remaining unselected links will be assigned with negative labels instead.

Table 7.1 Features extracted from user pair u and v from the homogeneous network

Features	Closeness measures
Common Neighbor (CN)	$\Gamma(u) \cap \Gamma(v)$
Jaccard's Coefficient (JC)	$\frac{\lvert\Gamma(u) \cap \Gamma(v)\rvert}{\lvert\Gamma(u) \cup \Gamma(v)\rvert}$
Adamic/Adar (AA)	$\sum_{w \in (\Gamma(u) \cap \Gamma(v))} \frac{1}{\log \lvert\Gamma(w)\rvert}$
Preferential Attachment (PA)	$\lvert\Gamma(u)\rvert \cdot \lvert\Gamma(v)\rvert$
Shortest Path (SP)	$\min\{\lvert p\rvert\}_{p \in \mathcal{P}_{u,v}}$
Katz	$\sum_{l=1}^{l_{max}} \beta^l \lvert\mathcal{P}^l_{u,v}\rvert$

7.2.1.3 Unsupervised Link Prediction Result Evaluation

Given all the remaining links in set $\mathcal{E}^r_{t_p,t_c}$ and the newly formed links in the future network G_{t_c,t_f}, we can represent their ground truth labels as vector $\mathbf{y} \in \{-1,+1\}^{|\mathcal{E}^r_{t_p,t_c}|\times 1}$. For all the links formed in the future network G_{t_c,t_f}, we can assign them with label $+1$, while for the remaining links, they will be assigned with label -1 instead.

Given the calculated scores, e.g., $\{C(e)\}_{e\in\mathcal{E}^r_{t_p,t_c}}$, and the ground truth label vector, we can evaluate the performance of the link prediction model by calculating the AUC score (i.e., the area under ROC curve). Among all the links, the top k links can be picked, the prediction result can also be evaluated with metrics like $nDCG@k$ [18].

Meanwhile, if the top k links are selected to assign with labels, the output of the link prediction model will be the prediction label of these links. By comparing them with the ground truth label vector, metrics like Precision, Recall, F1, and Accuracy (at top k) can be calculated as the performance evaluation results.

7.2.2 Supervised Link Prediction

In some cases, links in the networks are explicitly categorized into different groups, like links denoting friends vs those representing enemies, friends (with connections) vs strangers (no connections). Given a set of labeled links, e.g., set $\mathcal{E}$, containing links belonging to different classes, the *supervised link prediction* problem [14] aims at building a supervised learning model to address the link prediction problem. The learned model will be applied to determine the labels of links in the test set. In this part, we still take the link formation problem as an example to illustrate the supervised link prediction model.

7.2.2.1 Supervised Link Prediction Problem Setting

Given the network structure $G = (\mathcal{V}, \mathcal{E})$ with the formed links in set $\mathcal{E}$, we can represent all the potential links among users in network G as set $\mathcal{L} = \mathcal{V} \times \mathcal{V} \setminus \{(u,u)\}_{u\in\mathcal{V}} \setminus \mathcal{E}$. These existing links can be labeled as the positive training set, while a subset of links in $\mathcal{L}$ are identified as the links will never be formed, which can be denoted as $\mathcal{L}_n \subset \mathcal{L}$ and labeled as the negative training set. These positively and negatively labeled links can be treated as the training set, i.e., $\mathcal{L}_{train} = \mathcal{E} \cup \mathcal{L}_n$, and the remaining links with unknown labels can be used as the testing set $\mathcal{L}_{test} = \mathcal{L} \setminus \mathcal{L}_n$. In the supervised link prediction problem, we aim at building a supervised classification/regression model with the training set $\mathcal{L}_{train}$ and apply the learned model to infer the label of links in the testing set $\mathcal{L}_{test}$.

7.2.2.2 Supervised Link Prediction Feature Extraction

To represent each of the social links, like link $l = (u, v) \in \mathcal{E}$ between nodes u and v, a set of features representing the characteristics of the link l as well as nodes u, v will be extracted in the model building. Normally, the features can be extracted for links in the prediction task can be divided into two categories:

- *Features of Nodes*: The characteristics of the nodes can be denoted by the measures introduced in Chap. 3.3, like these various node centrality measures. For instance, for the link (u, v), based on the known links in the training set, we can compute the centrality measures based on degree, normalized degree, eigen-vector, Katz, PageRank, Betweenness of nodes u and v as part of the features for link (u, v), which can be denoted as vectors $\mathbf{x}_u$ and $\mathbf{x}_v$ respectively.

- *Features of Links*: The characteristics of the links in the networks can be calculated by computing the closeness between the nodes composing the nodes. For instance, for link (u, v), based on the known links in the training set, we can compute the closeness measures based on common neighbor, Jaccard's coefficient, Adamic/Adar, shortest path, Katz, hitting time, commute time, etc. between nodes u and v as the features for link (u, v), which can be denoted as vector $\mathbf{x}_{u,v}$ formally.

We can append the features for nodes u, v and those for link (u, v) together and represent the extracted feature vector for link $l = (u, v)$ as vector $\mathbf{x}_l = [\mathbf{x}_u^\top, \mathbf{x}_v^\top, \mathbf{x}_{u,v}^\top]^\top \in \mathbb{R}^k$ of length k.

7.2.2.3 Supervised Link Prediction Model

With the training set $\mathcal{L}_{train}$, we can represent the feature vectors and labels for the links in $\mathcal{L}_{train}$ as the training data $\{(\mathbf{x}_l, y_l)\}_{l \in \mathcal{L}_{train}}$. Meanwhile, with the testing set $\mathcal{L}_{test}$, we can represent the features extracted for the links in it as $\{\mathbf{x}_l\}_{l \in \mathcal{L}_{train}}$. Different classification models can be used as the base model for the link prediction task, like the decision tree model, artificial neural network model, and support vector machine (SVM) model introduced in Sect. 2.3. These models can be trained with the training data, and the labels of links in the testing set can be determined by applying models to the testing data instances.

Depending on the specific models being applied, the output of the link prediction result can include (1) the predicted labels of the links in $\mathcal{L}_{test}$, and (2) the prediction confidence scores/probability scores of links in $\mathcal{L}_{test}$.

7.2.2.4 Supervised Link Prediction Result Evaluation

Different evaluation metrics can be used for measuring the performance of the link prediction models. For the models producing the prediction labels of the test set, evaluation metrics like precision, recall, F1, and accuracy can be used in performance evaluation. Meanwhile, for the models producing the confidence score list as the output, evaluation metrics like AUC, and Precision@k, nDCG@k can be used in performance evaluation. For these metrics aforementioned, higher evaluation scores will correspond to better link prediction performance.

7.2.3 Matrix Factorization Based Link Prediction

Besides unsupervised and supervised link prediction models, many other methods based on matrix factorization can also be applied to solve the link prediction task in homogeneous networks [1,10,42].

7.2.3.1 Matrix Factorization Based Link Prediction Problem Setting

Given a homogeneous social network $G = (\mathcal{V}, \mathcal{E})$ and the existing social links among users in set $\mathcal{E}$, the remaining potential links among users can be represented as $\mathcal{L} = \mathcal{V} \times \mathcal{V} \setminus \{(u, u)\}_{u \in \mathcal{V}} \setminus \mathcal{E}$. The links in set $\mathcal{E}$ are the formed links and can be labeled as the positive instances, while those in set $\mathcal{L}$ contain both the links to be formed and those will never be formed (i.e., involve both positive and negative links) and should be unlabeled.

The training set available involves both the positively labeled links in set $\mathcal{E}$ and the unlabeled links in set $\mathcal{L}$. The testing set is the unlabeled set $\mathcal{L}$, and we aim at inferring the labels of these potential links with a matrix factorization based approach.

7.2.3.2 Matrix Factorization Based Link Prediction Model

Formally, given the homogeneous social network $G = (\mathcal{V}, \mathcal{E})$ and the existing social links among users in set $\mathcal{E}$, we can organize these links into the social adjacency matrix $\mathbf{A} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$. Given the adjacency matrix $\mathbf{A}$ of network G, a low-rank compact representation matrix, $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}| \times d}$, $d < |\mathcal{V}|$, can be used to store the social information for each user in the network. Matrix $\mathbf{U}$ can be obtained by solving the following optimization objective function:

$$\min_{\mathbf{U},\mathbf{V}} \left\| \mathbf{A} - \mathbf{U}\mathbf{V}\mathbf{U}^\top \right\|_F^2, \tag{7.1}$$

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

To avoid overfitting, regularization terms $\|\mathbf{U}\|_F^2$ and $\|\mathbf{V}\|_F^2$ are added to the object function as follows [42]:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \left\| \mathbf{A} - \mathbf{U}\mathbf{V}\mathbf{U}^\top \right\|_F^2 + \alpha \cdot \|\mathbf{U}\|_F^2 + \beta \cdot \|\mathbf{V}\|_F^2, \\ &s.t. \mathbf{U} \geq \mathbf{0}, \mathbf{V} \geq \mathbf{0}, \end{aligned} \tag{7.2}$$

where α and β are the weights 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:

$$\begin{aligned} \mathcal{F} = & Tr(\mathbf{A}\mathbf{A}^\top) - Tr(\mathbf{A}\mathbf{U}\mathbf{V}^\top\mathbf{U}^\top) \\ & - Tr(\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{A}^\top) + Tr(\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top\mathbf{U}^\top) \\ & + \alpha Tr(\mathbf{U}\mathbf{U}^\top) + \beta Tr(\mathbf{V}\mathbf{V}^\top) - Tr(\Theta\mathbf{U}) - Tr(\Omega\mathbf{V}) \end{aligned} \tag{7.3}$$

where Θ and Ω are the multipliers for the constraints on $\mathbf{U}$ and $\mathbf{V}$ respectively.

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

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \mathbf{U}} = & - 2\mathbf{A}^\top\mathbf{U}\mathbf{V} - 2\mathbf{A}\mathbf{U}\mathbf{V}^\top + 2\mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top \\ & + 2\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top + 2\alpha\mathbf{U} - \Theta^\top \end{aligned} \tag{7.4}$$

$$\frac{\partial \mathcal{F}}{\partial \mathbf{V}} = - 2\mathbf{U}^\top\mathbf{A}\mathbf{U} + 2\mathbf{U}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U} + 2\beta\mathbf{V} - \Omega^\top \tag{7.5}$$

By making $\frac{\partial \mathcal{F}}{\partial \mathbf{U}} = \mathbf{0}$ and $\frac{\partial \mathcal{F}}{\partial \mathbf{V}} = \mathbf{0}$ and using the KKT complementary condition, we can get:

$$\mathbf{U}(i,j) \leftarrow \mathbf{U}(i,j)\sqrt{\frac{\left(\mathbf{A}^\top\mathbf{U}\mathbf{V} + \mathbf{A}\mathbf{U}\mathbf{V}^\top\right)(i,j)}{\left(\mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V} + \mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top + \alpha\mathbf{U}\right)(i,j)}}, \tag{7.6}$$

$$\mathbf{V}(i,j) \leftarrow \mathbf{V}(i,j)\sqrt{\frac{\left(\mathbf{U}^\top\mathbf{A}\mathbf{U}\right)(i,j)}{\left(\mathbf{U}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U} + \beta\mathbf{V}\right)(i,j)}}. \tag{7.7}$$

The low-rank matrix $\mathbf{U}$ captures the information of each user from the adjacency matrix. The matrix $\mathbf{U}$ can be used in different ways. For instance, 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. Meanwhile, based on the matrix $\mathbf{V}$ learnt from the model, we can also represent the predicted score of link (u, v) as $\mathbf{U}_u \mathbf{V} \mathbf{U}_v^\top$, where notations $\mathbf{U}_u$ and $\mathbf{U}_v$ represent the rows in matrix $\mathbf{U}$ corresponding to users u and v, respectively.

7.2.3.3 Matrix Factorization Based Link Prediction Result Evaluation

Given the ground-truth labels of links in the unlabeled set $\mathcal{L}$ and their corresponding inferred scores based on the matrices $\mathbf{U}$ and $\mathbf{V}$ learnt from the model, we can evaluate the performance of the model with metrics like AUC and Precision@k as well as nDCG@k. In the exercise at the end of this chapter, we will ask the readers to try to implement the above link prediction algorithms with a preferred programming language, and compare their performance in inferring the social links within a homogeneous network.

7.3 Heterogeneous Network Collective Link Prediction

Homogeneous networks with one single type of nodes/links is a very simple network representation. In the real-world online social networks, there usually exist many different kinds of nodes, like *users, offline POIs, posts*. Users can also perform various kinds of actions, like *follow other users* and *check-in at some places*, which will create very complex connections among these nodes. Formally, for the online social networks with such a complex structure, they are called the heterogeneous information networks. There exist very diverse online social networks in the real world. In this section, we will be mainly focused on the online social networks providing the geographic services, which are called the location based social networks (LBSNs) [7], and study the *collective link prediction* task based on the LBSNs [58].

7.3.1 Introduction to LBSNs

Location-based social networks (LBSNs) are one kind of online social networks that can provide geographic services, e.g., location check-ins and posting reviews, and have been attracting much attention in recent years [7, 33, 45, 48, 49]. LBSNs usually have very complex structures, including multiple kinds of nodes (e.g., users, locations, etc.) and different types of links among these nodes (e.g., social links among users and location links between users and locations). For example, Foursquare[1] is a mainstream LBSN. It involves millions of users and locations. Foursquare users can add friends, check-in at different locations with their mobile phones, write reviews, and share the locations with their friends.

Many important services offered by LBSNs can be cast as the link prediction problems. For example, friend recommendation involves predicting social links among users; location recommendation aims at predicting location links between users and locations. LBSNs can benefit a lot from the high-quality social link and location link prediction results. The reason is that well-established social ties can improve user's engagement in social networks [21]. Meanwhile, in location-based social networks, high-quality predicted location links can enhance the value of the location services in the networks.

Conventional link prediction researches on LBSNs mostly focus on predicting either social links [33, 45] or location links [7, 48] and usually assume that the prediction tasks of different types of links to be independent. However, in many real-world LBSNs, the link prediction tasks for social links

[1] https://foursquare.com.

and location links are strongly correlated and mutually influential to each other [58]. For example, if two users are friends with each other, they are more likely to check-in at similar locations. Thus the performance of location recommendation can be significantly improved if we could make accurate friendship predictions. Similarly, if two users often check-in at similar locations, they are more likely to know each other and make friends in the real life. Viewed in this perspective, the performance of friend recommendation can be greatly improved if we could make accurate location-link predictions.

7.3.2 Collective Link Prediction

In this section, we study the collective link prediction problem for LBSNs as introduced in [58] and the links to be predicted include both social links and location links. The problem is very challenging to solve due to the fact that social links and location links in LBSNs are correlated instead of being independent. The prediction tasks on social links and location links should be considered at the same time. Many existing works mainly focus on predicting one single type of links in LBSNs [7,33,45,48], which fail to consider the correlations between different link prediction tasks.

In the following part, we will introduce a supervised collective linkage transferring method, TRAIL (TRAnsfer heterogeneous lInks across LBSNs), proposed in [58] to address the above challenges. TRAIL can accumulate auxiliary information for locations from online posts which have check-ins at them and can extract heterogeneous features for both social links and location links. TRAIL can predict social links and location links simultaneously.

Let $G = (\mathcal{V}, \mathcal{E})$ be the networks studied in this section, where $\mathcal{V} = \bigcup_i \mathcal{V}_i$ is the union of different types of nodes and $\mathcal{V}_i, i \in \{1, 2, \ldots, \}$ is the set of nodes of the i_{th} type. $\mathcal{E} = \bigcup_j \mathcal{E}_j$ is the union of link sets among nodes in $\mathcal{V}$ and $\mathcal{E}_j, j \in \{1, 2, \ldots\}$ is the set of links of the j_{th} type. Specially, for a LBSN, node set $\mathcal{V} = \mathcal{U} \cup \mathcal{L} \cup \mathcal{T} \cup \mathcal{W}$ is the union of node sets of users, locations, time, and words. The link set $\mathcal{E} = \mathcal{E}_s \cup \mathcal{E}_l \cup \mathcal{E}_t \cup \mathcal{E}_w$ is the union of link sets consisting of social friendships links, and the links between users with location check-ins, active time, and published words, respectively.

Given an LBSN $G = (\mathcal{V}, \mathcal{E})$ with the existing social links $\mathcal{E}_s$ and location links $\mathcal{E}_l$, what we want to predict in the studied network are a subset of potential social links among users in G: $\mathcal{L}_s \subset (\mathcal{U} \times \mathcal{U} \setminus \mathcal{E}_s)$ and a subset of potential location check-in links in G: $\mathcal{L}_l \subset (\mathcal{U} \times \mathcal{L} \setminus \mathcal{E}_l)$. In other words, we want to build a mapping: $f : \{\mathcal{L}_s, \mathcal{L}_l\} \to \{-1, 1\}$ to decide whether potential links in $\{\mathcal{L}_s, \mathcal{L}_l\}$ exist or not and a confidence score function $P : \{\mathcal{L}_s, \mathcal{L}_l\} \to [0, 1]$ denoting their existence probabilities. In the following parts, we will introduce the supervised collective link transferring method, TRAIL, in detail to address the problem.

7.3.3 Information Accumulation and Feature Extraction

TRAIL is based on a supervised learning setting and, as a result, we need to extract features for both social links and location links using the heterogeneous information in the network. Slightly different from users, the locations in online social networks cannot generate information on their own. Before introducing the extracted features, we will introduce a method to accumulate information for locations at first.

7.3.3.1 Information Accumulation for Locations

Locations are represented as $(latitude, longitude)$ pairs in the studied problem, which possess no auxiliary information except location links with users in the network. As a result, we will confront problems of lacking auxiliary information when extracting heterogeneous features for location links

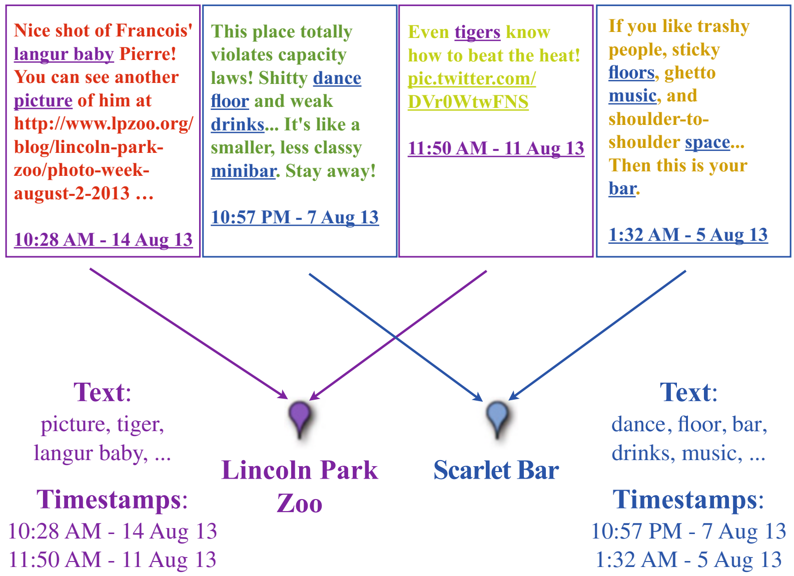


Fig. 7.1 Example of information accumulation for locations from online posts (both Lincoln Park and Scarlet Bar are located in Chicago, US)

between users and locations. Actually, we notice that users can publish online posts at the locations, and the textual contents and timestamps information of the online posts checked in at a certain location can be accumulated as the auxiliary information possessed by that location.

From a statistical point of view, information from posts published at a certain location, including both timestamps and text contents, can reveal some properties of the location. For example, the timestamps of most posts published at nightlife sites are after 6:00 PM. While those of posts published at restaurants serving brunch are during the daytime. Posts published at national parks can contain some phrases depicting the scenes, while posts published at basketball court may be mostly talking about games, teams, and players. So, we can know more about the locations from the information accumulated from online posts.

Example 7.1 For example, in Fig. 7.1, we have two totally different locations: the Lincoln Park Zoo[2] and Scarlet Bar.[3] The Lincoln Park Zoo is the largest free zoo in Chicago and is open during 10:00 AM–5:00 PM. The Scarlet Bar is one of the most famous bars in Chicago, where people can drink with friends, dance to enjoy their night life, and it is open during 8:00 PM–2:00 AM.

We also have 4 online posts published by people at these two places in either Foursquare or Twitter. From the contents of these posts, we find that people usually publish words about animals, pictures, and the scene at the Lincoln Park Zoo. However, people who visit the Scarlet Bar mainly talk about the atmosphere in the bar, the drinks, the dance floor, and the music there. So, users who frequently talk about animals in daily life can be interested in the Lincoln Park Zoo, while those who usually post words about the drinks may like the Scarlet Bar more. Meanwhile, we can also accumulate the timestamps of posts published at these two places. The timestamps of posts published at the Lincoln Park Zoo are mostly during the daytime, while those of posts published at the Scarlet Bar are at night. So, users who are usually active in the daytime can be more likely to visit the Lincoln Park Zoo, while people who are active during the night may prefer the Bar.

[2] http://www.lpzoo.org.

[3] http://www.scarletbarchicago.com.

Table 7.2 Features extracted from vector $\boldsymbol{x}$ and $\boldsymbol{y}$

Features	Descriptions
Extended Degree Count (EDC)	$\|\|\boldsymbol{x}\|\|_1, \|\|\boldsymbol{y}\|\|_1$
Extended Degree Ratio (EDR)	$\|\|\boldsymbol{x}\|\|_1 / \|\|\boldsymbol{y}\|\|_1$
Extended Common Neighbor (ECN)	$\boldsymbol{x} \cdot \boldsymbol{y}$
Extended Jaccard's Coefficient (EJC)	$\frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\\|\boldsymbol{x}\\|_1 \cdot \\|\boldsymbol{y}\\|_1}$
Extended Preferential Attachment (EPA)	$\|\|\boldsymbol{x}\|\|_1 \cdot \|\|\boldsymbol{y}\|\|_1$
Euclidean Distance (ED)	$(\sum_k (x_k - y_k)^2)^{1/2}$
Cosine Similarity (CS)	$\frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\\|\boldsymbol{x}\\|_2 + \\|\boldsymbol{y}\\|_2}$

7.3.3.2 Heterogeneous Features

Based on the heterogeneous information in the networks, we will extract 4 different categories of features for both social links and location links from the heterogeneous information in the network, which include *social features*, *spatial distribution features*, *text usage features*, and *temporal distribution features*. A summary of frequently used features is available in Table 7.2, where $||\mathbf{x}||_p = (\sum_{i=1}^{|\mathbf{x}|} |x_i|^p)^{1/p}$ denotes the L_p-norm of vector $\mathbf{x}$.

- **Features of Social Links:** For a certain social link (u_i, u_j), we can get their neighbors from the network, which can be represented as sets $\Gamma(u_i)$ and $\Gamma(u_j)$, respectively. Based on $\Gamma(u_i)$, we can construct the social link weight vector $\tilde{s}(u_i)$ for u_i, where $\tilde{s}(u_i) = (p_{1,i}, p_{2,i}, \ldots, p_{k,i}, \ldots, p_{n,i})^\top$ and $n = |\mathcal{U}|$ is the size of user set and $p_{k,i}$ is the weight of social link $(u_k, u_i), \forall u_k \in \mathcal{U}$: if $u_k \in (\mathcal{U} \setminus \Gamma(u_i))$, $p_{k,i} = 0.0$; if $u_k \in \Gamma(u_i)$ and link (u_k, u_i) exists originally, then $p_{k,i} = 1.0$; otherwise, $p_{k,i}$ is the existence probability of link (u_k, u_i). Similarly, we can construct vector $\tilde{s}(u_j)$ for user u_j, which is of the same length as $\tilde{s}(u_i)$. From $\tilde{s}(u_i)$ and $\tilde{s}(u_j)$, 7 different *social features* are extracted for social link (u_i, u_j), which are summarized in Table 7.2.

  In a similar way, for a certain social link (u_i, u_j), we can get the set of locations visited by user u_i and u_j as sets $\Phi(u_i)$ and $\Phi(u_j)$, from which we can obtain their location link weight vectors as $\tilde{l}(u_i)$ and $\tilde{l}(u_j)$, where the entries denote the times that these users visit the locations. From the timestamps of posts published by users, we can obtain the users' active patterns. Each day is divided into 24 slots and the ratio of online posts published by user u in each hour is saved in a temporal distribution vector $\tilde{t}(u)$, whose length is 24. For social link (u_i, u_j), we can construct the temporal distribution vectors: $\tilde{t}(u_i)$ and $\tilde{t}(u_j)$ for u_i and u_j. In addition, we transform the words used by two users u_i and u_j into two text usage vectors: $\tilde{w}(u_i)$ and $\tilde{w}(u_j)$ weighted by TF-IDF [31], which are of the same length. From these vectors, we can extract the *spatial distribution features*, *temporal distribution features*, and *text usage features* similar to the social link features summarized in Table 7.2 for social link (u_i, u_j).
- **Features of Location Links:** Similarly, we can obtain the set of users who have visited a location and regard them as the "neighbors" of that location. And for a location link (u_i, l_j), we can get the sets of neighbors of u_i and l_j as $\Gamma(u_i)$ and $\Psi(l_j)$, from which we can construct the social link weight vectors $\tilde{s}(u_i)$ and $\tilde{s}(l_j)$, respectively. From the accumulated text and timestamps information of locations and the auxiliary information owned by users, we can also construct the temporal distribution vectors $\tilde{t}(u_i)$ and $\tilde{t}(l_j)$ and the text usage vectors $\tilde{w}(u_i)$ and $\tilde{w}(l_j)$ for location link (u_i, l_j). From these vectors, we can extract the *social features*, *temporal distribution features*, and *text usage features* for location link (u_i, l_j).

  In addition, according to previous definitions, we can get the locations that user u has visited in the past: $\Phi(u)$ and the location link weight vector $\tilde{l}(u)$ of u as well as the neighbors $\Psi(l)$ of a location l and its social link weight vector: $\tilde{s}(l)$. For a certain location link (u_i, l_j), we extract 3 spatial distribution features for the location links from the network:

(1) average weighted geographic distance between locations in $\Phi(u_i)$ and l_j

$$\frac{\sum_{l_k \in \Phi(u_i)} GeoD(l_k, l_j) \cdot \tilde{l}(u_i)_{l_k}}{||\tilde{l}(u_i)||_1 \cdot |\Phi(u_i)|}, \tag{7.8}$$

where $GeoD(l_k, l_j)$ is the geographic distance (e.g., the manhattan distance [2]) between l_k and l_j and $\tilde{l}(u_i)_{l_k}$ is the weight of location link (u_i, l_k) saved in u_i's location link weight vector.

(2) weighted number of users who have visited both locations in $\Phi(u_i)$ and l_j

$$\sum_{l_k \in \Phi(u_i)} \tilde{s}(l_k) \cdot \tilde{s}(l_j) \cdot \tilde{l}(u_i)_{l_k} \tag{7.9}$$

(3) average weighted number of users who have visited both locations in $\Phi(u_i)$ and l_j

$$\frac{\sum_{l_k \in \Phi(u_i)} \tilde{s}(l_k) \cdot \tilde{s}(l_j) \cdot \tilde{l}(u_i)_{l_k}}{||\tilde{l}(u_i)||_1 \cdot \sum_{l_k \in \Phi(u_i)} ||\tilde{s}(l_k)||_1} \tag{7.10}$$

7.3.4 Collective Link Prediction Model

In this section, we will analyze and formulate the correlations between the social link prediction task and the location link prediction task, and introduce an integrated collective link prediction framework to address both of these two tasks simultaneously.

7.3.4.1 Correlation Between Different Tasks

When predicting a link with the supervised link prediction models introduced before, the classifiers will give a score within range [0, 1] to show its existence probability. Newly predicted social links will update the social link existence probability information in the network, which can affect the prediction of other location links. For example, these updated social link existence probabilities can change the extended common neighbors of a location and a user, and may further change the prediction results. Similarly, the location link prediction task can also influence the social link prediction result.

Example 7.2 For example, in Fig. 7.2, we show an example of different link prediction methods. Figure 7.2a 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-check-ins), and other auxiliary information, traditional link prediction methods can predict social links and locate links independently. Figure 7.2b shows the independent social link prediction result, in which social links (u_2, u_3) and (u_1, u_3) are predicted to be positive (i.e., existing), while the other two social links (u_1, u_2) and (u_2, u_4) are predicted to be negative (i.e., non-existing). Figure 7.2c 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 positive (i.e., existing), while (u_2, l_3) and (u_3, l_3) is predicted to be negative (i.e., non-existing).

From the results in Fig. 7.2b, c, we can find some problematic phenomena. For example, user u_2 and u_1 are predicted that they will visit locations l_1, l_2 and they are also predicted to share a common

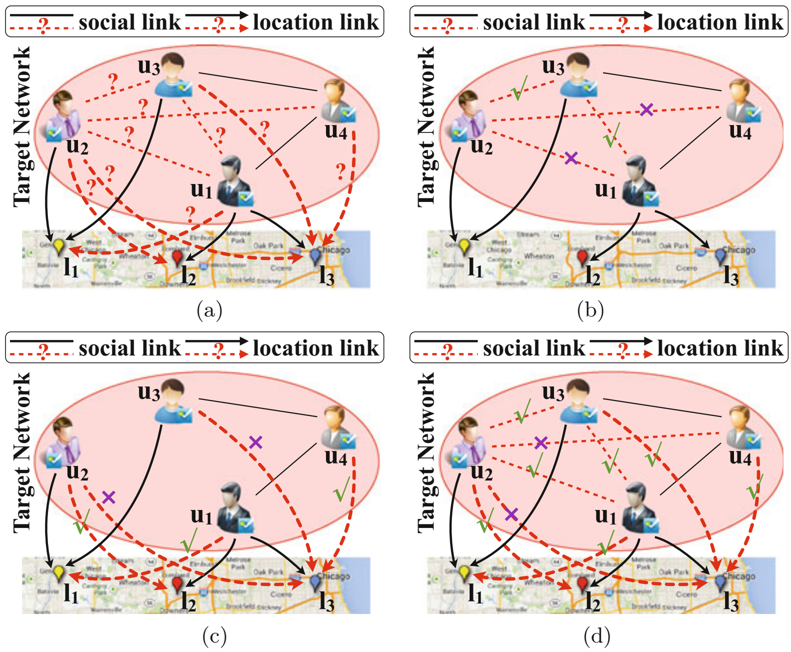


Fig. 7.2 An example of different link prediction methods. (**a**) The input network. (**b**), (**c**) Independent social link and location link prediction result. (**d**) The collective link prediction result

neighbor: u_3. Based on the result, it is highly likely that these two users may know each other, and the potential social link (u_2, u_3) will be predicted to be existing. However, according to the independent prediction result, it is predicted to be non-existing as shown in Fig. 7.2b. Another example is that many neighbors of user u_3, including both the originally existing u_4 and the newly predicted u_1, have visited or are predicted to have visited l_3. Based on such an observation, 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 non-existing in Fig. 7.2c.

If we consider the correlation between these two link prediction tasks 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 existing. In Fig. 7.2d, we show a potential result of collective link prediction methods, where the prediction results of social links and location links seem to be much more consistent.

7.3.4.2 Collective Link Prediction

As introduced before, we represent the sets of potential social links and potential location links to be predicted as $\mathcal{L}_s \subset (\mathcal{U} \times \mathcal{U} \setminus \mathcal{E}_s)$ and $\mathcal{L}_l \subset (\mathcal{U} \times \mathcal{L} \setminus \mathcal{E}_l)$, respectively, in the problem formulation section. For links $l_s \in \mathcal{L}_s$ and $l_l \in \mathcal{L}_l$, the supervised models built with the existing information in the network will give them the predicted labels: $y(l_s)$ and $y(l_l)$, as well as the existence probability scores: $P(y(l_s) = 1)$ and $P(y(l_l) = 1)$. Traditional methods predicting social links and location links independently aim at finding the set of labels achieving the maximum likelihood scores for each kind of these links. In other words, let $\hat{\mathcal{Y}}_s \subset \{-1, 1\}^{|\mathbf{L}_s|}$, $\hat{\mathcal{Y}}_l \subset \{-1, 1\}^{|\mathbf{L}_l|}$ be the sets of optimal labels, the objective functions of the social and location link prediction tasks can be denoted as

$$\hat{\mathcal{Y}}_s = arg \max_{\mathcal{Y}_s} P(y(\mathcal{L}_s) = \mathcal{Y}_s | \mathbf{x}(\mathcal{L}_s)), \tag{7.11}$$

$$\hat{\mathcal{Y}}_l = arg \max_{\mathcal{Y}_l} P(y(\mathcal{L}_l) = \mathcal{Y}_l | \mathbf{x}(\mathcal{L}_l)), \tag{7.12}$$

where $P(y(\mathcal{L}_s) = \mathcal{Y}_s)$ and $P(y(\mathcal{L}_l) = \mathcal{Y}_l)$ 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 be incorporated into the same framework. The jointly optimal label sets $\hat{\mathcal{Y}}_s$ and $\hat{\mathcal{Y}}_l$ will be

$$\hat{\mathcal{Y}}_s, \hat{\mathcal{Y}}_l = arg \max_{\mathcal{Y}_s, \mathcal{Y}_l} P(y(\mathcal{L}_s) = \mathcal{Y}_s | y(\mathcal{L}_l) = \mathcal{Y}_l, \mathbf{x}(\mathcal{L}_s)) \times P(y(\mathcal{L}_l) = \mathcal{Y}_l | y(\mathcal{L}_s) = \mathcal{Y}_s, \mathbf{x}(\mathcal{L}_l)) \tag{7.13}$$

For the given optimization equation, there are many different solutions. In this part, we will give an iterative method, TRAIL, to address 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 we have

$$\hat{\mathcal{Y}}_s^{(\tau)} = arg \max_{\mathcal{Y}_s} P(y(\mathcal{L}_s) = \mathcal{Y}_s | G, y(\mathcal{L}_s) = \hat{\mathcal{Y}}_s^{(\tau-1)}, y(\mathcal{L}_l) = \hat{\mathcal{Y}}_l^{(\tau-1)}) \tag{7.14}$$

$$\hat{\mathcal{Y}}_l^{(\tau)} = arg \max_{\mathcal{Y}_l} P(y(\mathcal{L}_l) = \mathcal{Y}_s | G, y(\mathcal{L}_s) = \hat{\mathcal{Y}}_s^{(\tau)}, y(\mathcal{L}_l) = \hat{\mathcal{Y}}_l^{(\tau-1)}). \tag{7.15}$$

The pseudo code of TRAIL is available in Algorithm 1. Here, we mainly focus on providing the overall framework of TRAIL and haven't specified the classifier models to be used. Actually, any classification algorithms (e.g., SVM, Neural Networks) we have introduced before can all be adopted as the base classifier in the framework.

Algorithm 1 TRAIL

Require: heterogeneous LBSN, G.
existing social links and location links: E_s, E_l
potential social links and location links: L_s, L_l
Ensure: the inferred labels and existence probabilities of links in L_s and L_l: $\hat{\mathcal{Y}}_s, \hat{\mathcal{P}}_s, \hat{\mathcal{Y}}_l, \hat{\mathcal{P}}_l$
construct training sets, test sets with E_s, E_l, L_s and L_l.
$converge \leftarrow False$
while $converge$ is $False$ **do**
  extract features $\mathbf{x}(E_s)$ and $\mathbf{x}(L_s)$ for social links in E_s and L_s from G.
  $C_s \leftarrow$ **train**$([\mathbf{x}(E_s)^T, \mathbf{x}^s(E_s)^T, y^s(E_s)]^T, y(E_s))$
  $\hat{\mathcal{Y}}_s, \hat{\mathcal{P}}_s \leftarrow C_s$.**classify**$([\mathbf{x}(L_s)^T, \mathbf{x}^s(L_s)^T, y^s(L_s)]^T)$
  update G with $\hat{\mathcal{Y}}_s, \hat{\mathcal{P}}_s$
  Accumulate information for locations
  extract features $\mathbf{x}(E_l)$ and $\mathbf{x}(L_l)$ for location links in E_l and L_l from G.
  $C_l \leftarrow$ **train**$([\mathbf{x}(E_l)^T, \mathbf{x}^s(E_l)^T, y^s(E_l)]^T, y(E_l))$
  $\hat{\mathcal{Y}}_l, \hat{\mathcal{P}}_l \leftarrow C_l$.**classify**$([\mathbf{x}(L_l)^T, \mathbf{x}^s(L_l)^T, y^s(L_l)]^T)$
  update G with $\hat{\mathcal{Y}}_l, \hat{\mathcal{P}}_l$
  if $\hat{\mathcal{Y}}_s, \hat{\mathcal{P}}_s, \hat{\mathcal{Y}}_l, \hat{\mathcal{P}}_l$ all converge **then**
    $converge \leftarrow True$
  end if
end while
Return $\hat{\mathcal{Y}}_s, \hat{\mathcal{P}}_s, \hat{\mathcal{Y}}_l, \hat{\mathcal{P}}_l$

7.4 Cold Start Link Prediction for New Users

In this section, we study the problem of predicting social links for new users, who have created their accounts for just a short period of time. Generally, new users who have just created the accounts, they are more likely to accept the recommendations to establish their social communities. However, the limited information available for these new users can pose a great challenge on high quality recommendations of friends. Meanwhile, for the users who are new in one network, they may have been involved in other online social networks for a long time. Information can be transferred from these mature source networks for these users to the target network that we are focused on to resolve the lack of information problem.

7.4.1 New User Link Prediction Problem Description

Many of previous works on link prediction [1, 13, 14, 23, 53] focus on predicting potential links that will appear among all the users, based upon a snapshot of the social network. These works treat all users equally and try to predict social links for all users in the network. However, in real-world social networks, many new users are joining in the online social networks every day. It has been shown in previous works that there is a negative correlation between the age of nodes in the network and their link attachment rates. Predicting social links for these new users are more important than for those existing active users in the network as it will leave a good first impression on the new users. First impression often has a lasting impact on a new user and may decide whether he/she will become an active user. A bad first impression can turn a new user away. So it is important to make meaningful recommendations to new users to create a good first impression and attract them to participate more. For simplicity, we refer users that have been actively using the network for a long time as "old users".

The link prediction problem for new users is different from traditional link prediction problems. Conventional supervised link prediction methods implicitly or explicitly assume that the information are identically distributed over all the nodes in the network without considering the joining time of the users. The models trained over one part of the network can be directly used to predict links in other parts of the network. However, in real-world social networks, the information distributions of the new users could be very different from that of old users. New users may have only a few activities or even no activities (i.e., no social links or other auxiliary information) in the network; while old users usually have abundant activities and auxiliary information in the network. In Figs. 7.3 and 7.4, we show the degree distributions of the new users who registered their accounts within 3 months and the old users who registered more than 3 months before in Twitter and Foursquare, respectively. In

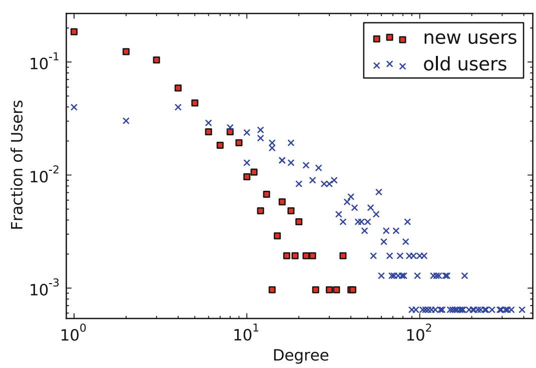


Fig. 7.3 Degree distributions of users in Foursquare network

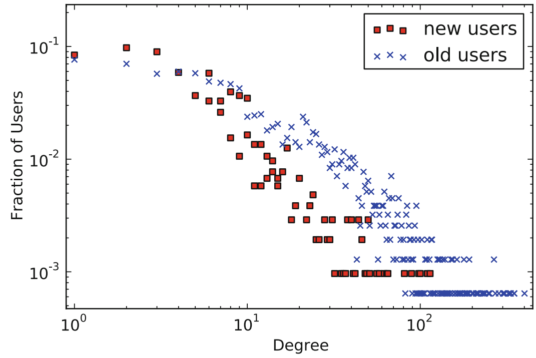


Fig. 7.4 Degree distributions of users in Twitter network

the plots, the x axis denotes the node degrees and the y axis denotes the fraction of users with certain degrees. We observe that the social link distributions of new users and old users are totally different from each other in both Foursquare and Twitter. As a result, conventional supervised link prediction models trained over old users based upon structural features, such as *common neighbors*, may not work well on the new users.

Another challenging problem in link prediction for new users is that information owned by new users can be very rare or even totally missing. Conventional methods based upon one single network will not work well due to the lack of historical data about the new users. In order to solve this problem, we need to transfer additional information about the new users from other sources. Nowadays, people are usually involved in multiple social networks to enjoy more services. For example, people will join Foursquare to search for nearby restaurants to have dinner with their family. Meanwhile, they tend to use Face book to socialize with their friends and involve in Twitter to post comments about recent news. The accounts of the same user in different networks can be linked through account alignments. For example, when users register their Foursquare accounts, they can use their Face book or Twitter accounts to sign in the Foursquare network. Such links among accounts of the same user are named as "anchor links" [19, 57–59] according to the description in Sect. 3.4.3, which could help align users' accounts across multiple social networks. For example, in Fig. 7.5, there are many users in two networks, respectively. We find that the accounts in these two networks are actually owned by 6 different users in reality and we add an *anchor link* between each pair of user accounts corresponding to the same user. Via the *anchor links*, we could locate users' corresponding accounts in the other networks.

New users in one social network (i.e., *target* network) might have been using other social networks (i.e., *source* networks) for a long time. These user accounts in the source networks can provide additional information about the new users in the source network. This additional information is crucial for link prediction about these new users, especially when the new users have little activities or no activities in the target network (i.e., cold start problem).

Example 7.3 For instance, in Fig. 7.5, we have two social networks, i.e., the target network and the source network, with aligned user accounts. In the target network, there are many old users with abundant social links and auxiliary information, such as posts, spatial and temporal activities. In addition, there are also some new users, i.e., user u_1^t and u_2^t, in the target network. These two new users have just created their accounts in the target network and have not yet created many social links or auxiliary information. However, we can see that there is abundant information about these two new users in the source network, based on their "anchor linked" user accounts u_1^s and u_2^s in the source

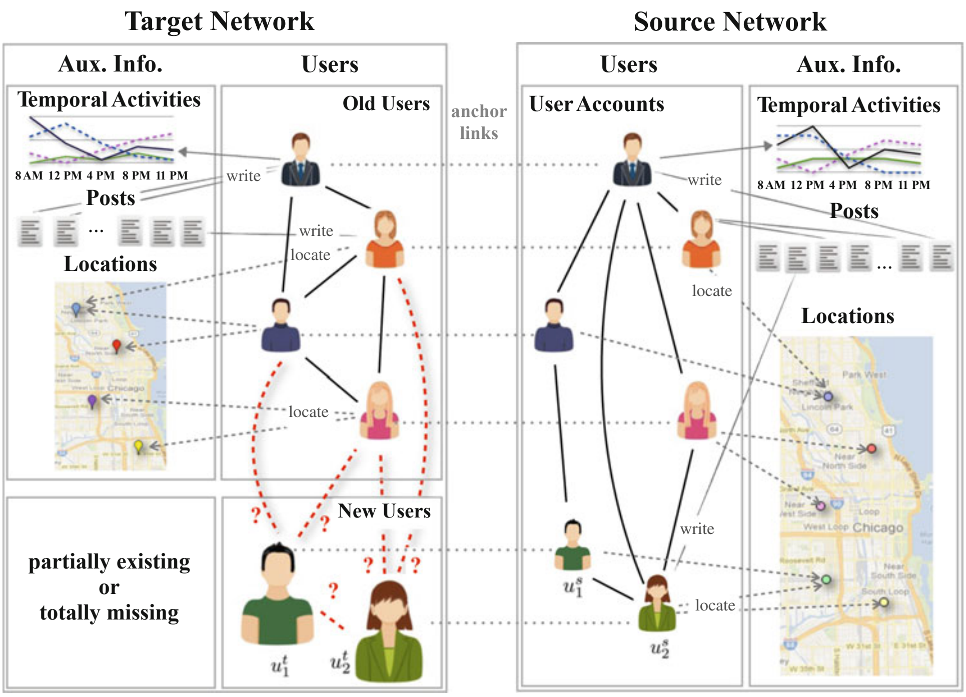


Fig. 7.5 Example of predicting social links across two aligned heterogeneous online social networks

network. The new users' information in source networks can be exploited to help improve the link prediction performances in the target network.

In order to solve these problems, in this section, we will introduce a novel supervised cross aligned networks link recommendation method, SCAN, proposed in [57]. Different from previous works, SCAN extracts heterogeneous features from other aligned networks to improve link prediction results for new users in the target network. SCAN analyzes the problem about the differences in information distributions between new users and old users in great detail and proposes a within-network personalized sampling method to accommodate that difference. What's more, SCAN can also solve the cold start social link prediction problem assisted by other aligned source networks. Intra- and inter-network information transfer can be conducted simultaneously in SCAN to make a full use of the information contained in these aligned networks to improve the prediction results.

7.4.2 Cold Start Link Prediction Problem Formulation

The problem studied in this section is social link prediction for new users. We will introduce a supervised method based on aligned heterogeneous networks. Let $\mathcal{G} = ((G^t, G^s), (\mathcal{A}^{t,s}))$ be two aligned heterogeneous social networks, where G^t is the target network and G^s is an aligned source network. $\mathcal{A}^{t,s}$ denotes the set of anchor links between G^t and G^s. We want to predict social links for the new users in the target network. Let $\mathcal{U}^t = \mathcal{U}^t_{new} \cup \mathcal{U}^t_{old}$ be the user set in G^t, where $\mathcal{U}^t_{new}$ and $\mathcal{U}^t_{old}$ are the sets of new users and old users, respectively, and $\mathcal{U}^t_{new} \cap \mathcal{U}^t_{old} = \emptyset$. What we want to predict is a subset of potential social links between the new users and all other users: $\mathcal{L} \subseteq \mathcal{U}^t_{new} \times \mathcal{U}^t$. In other words, we want to build a function $f : \mathcal{L} \to \{0, 1\}$, which could decide whether certain links related to new users exist in the target network or not.

7.4.3 Link Prediction Within Target Network

Based on the heterogeneous information available in the online social networks, a set of features can be extracted for the social links as introduced in Sect. 7.3.3.2. Next we will introduce how to use these features to build supervised methods to predict links for new users in the target network. Before doing that, we notice that the new users' information distribution can be totally different from that of the old users in the target network. However, information of both new users and old users is so important that should be utilized. In this section, we will analyze the differences in information distributions of new users and old users in the target network and propose a personalized within-network sampling method to process old users' information to accommodate the differences. Then, we will extend the traditional supervised link prediction method by using the old users' sampled information in the target network to improve the prediction results.

7.4.3.1 Sampling Old Users' Information

A natural challenge inherent in the usage of the target network to predict social links for new users is the differences in information distributions of new users and old users as mentioned before. To address this problem, the SCAN model proposes to accommodate old users' and new users' sub-networks by using a within-network personalized sampling method to process old users' information. Totally different from the link prediction with sampling problem studied in [3], SCAN conducts personalized sampling within the target network, which contains heterogeneous information, rather across multiple non-aligned homogeneous networks. And the link prediction target are the new users in the target network. By sampling the old users' sub-network, we want to achieve the following objectives:

- *Maximizing Relevance*: We aim at maximizing the relevance of the old users' sub-network and the new users' sub-network to accommodate differences in information distributions of new users and old users in the heterogeneous target network.
- *Information Diversity*: Diversity of old users' information after sampling is still of great significance and should be preserved.
- *Structure Maintenance*: Some old users possessing sparse social links should have higher probability to survive after sampling to maintain their links so as to maintain the network structure.

Let the heterogeneous target network be $G^t = \{\mathcal{V}^t, \mathcal{E}^t\}$, and $\mathcal{U}^t = \mathcal{U}^t_{old} \cup \mathcal{U}^t_{new} \subset \mathcal{V}^t$ is the set of user nodes (i.e., set of old users and new users) in the target network. Personalized sampling is conducted on the old users' part: $G^t_{old} = \{\mathcal{V}^t_{old}, \mathcal{E}^t_{old}\}$, in which each node is sampled independently with the sampling rate distribution vector $\boldsymbol{\delta} = (\delta_1, \delta_2, \ldots, \delta_n)$, where $n = \left|\mathcal{U}^t_{old}\right|$, $\sum_{i=1}^{n} \delta_i = 1$ and $\delta_i \geq 0$. Old users' heterogeneous sub-network after sampling is denoted as $\bar{G}^t_{old} = \{\bar{\mathcal{V}}^t_{old}, \bar{\mathcal{E}}^t_{old}\}$.

The main objective of the old users' information sampling is to make the old users' sub-network as relevant to new users' as possible. To measure the similarity score of a user u_i and a heterogeneous network G, we define a relevance function as follows:

$$R(u_i, G) = \frac{1}{|\mathcal{U}|} \sum_{u_j \in \mathcal{U}} S(u_i, u_j) \tag{7.16}$$

where $\mathcal{U}$ is the user set of network G and $S(u_i, u_j)$ measures the similarity between user u_i and u_j in the network. Each user has social relationships as well as other auxiliary information and $S(u_i, u_j)$ is defined as the average of similarity scores of these two parts:

$$S(u_i, u_j) = \frac{1}{2}(S_{aux}(u_i, u_j) + S_{social}(u_i, u_j)) \tag{7.17}$$

In our problem settings, the auxiliary information of each users could also be divided into 3 categories: *location*, *temporal*, and *text*. So, $S_{aux}(u_i, u_j)$ is defined as the mean of these three aspects.

$$S_{aux}(u_i, u_j) = \frac{1}{3}(S_{text}(u_i, u_j) + S_{loc}(u_i, u_j) + S_{temp}(u_i, u_j)) \tag{7.18}$$

There are many different methods measuring the similarities of these auxiliary information in different aspects, e.g. cosine similarity [16, 53]. As to the social similarity, Jaccard's coefficient [17] can be used to depict how similar two users are in their social relationships. We will not talk about these measures in this part.

The relevance between the sampled old users' network and the new users' network could be defined as the expectation value of function $R(\bar{u}^t_{old}, G^t_{new})$:

$$\begin{aligned} R(\bar{G}^t_{old}, G^t_{new}) &= \mathbb{E}(R(\bar{u}^t_{old}, G^t_{new})) \\ &= \frac{1}{|\mathcal{U}^t_{new}|} \sum_{j=1}^{|\mathcal{U}^t_{new}|} \mathbb{E}(S(\bar{u}^t_{old}, u^t_{new,j})) \\ &= \frac{1}{|\mathcal{U}^t_{new}|} \sum_{j=1}^{|\mathcal{U}^t_{new}|} \sum_{i=1}^{|\mathcal{U}^t_{old}|} \delta_i \cdot S(\bar{u}^t_{old,i}, u^t_{new,j}) \\ &= \boldsymbol{\delta}^\top \mathbf{s} \end{aligned} \tag{7.19}$$

where vector $\mathbf{s}$ equals:

$$\frac{1}{|\mathcal{U}^t_{new}|}[\sum_{j=1}^{|\mathcal{U}^t_{new}|} S(\bar{u}^t_{old,1}, u^t_{new,j}), \ldots, \sum_{j=1}^{|\mathcal{U}^t_{new}|} S(\bar{u}^t_{old,n}, u^t_{new,j})]^\top \tag{7.20}$$

and $\left|\mathcal{U}^t_{old}\right| = n$. Besides the relevance, we also need to ensure that the diversity of information in the sampled old users' sub-network could be preserved. Similarly, it also includes diversities of the auxiliary information and social relationships. The diversity of auxiliary information is determined by the sampling rate δ_i, which could be defined with the averaged *Simpson Index* [36] over the old users' sub-network.

$$D_{aux}(\bar{G}^t_{old}) = \frac{1}{\left|\mathcal{U}^t_{old}\right|} \cdot \sum_{i=1}^{|\mathcal{U}^t_{old}|} \delta_i^2 \tag{7.21}$$

As to the diversity in the social relationship, we could get the existence probability of a certain social link (u_i, u_j) after sampling to be proportional to $\delta_i \cdot \delta_j$. So, the diversity of social links in the sampled network could be defined as average existence probabilities of all the links in the old users' sub-network.

$$D_{social}(\bar{G}^t_{old}) = \frac{1}{\left|\mathcal{E}^t_{old,s}\right|} \cdot \sum_{i=1}^{|\mathcal{U}^t_{old}|} \sum_{j=1}^{|\mathcal{U}^t_{old}|} \delta_i \cdot \delta_j \times \mathbb{I}(u_i, u_j) \tag{7.22}$$

where $\left|\mathcal{E}^t_{old,s}\right|$ is the size of social link set of old users' sub-network and $\mathbb{I}(u_i, u_j)$ is an indicator function $\mathbb{I} : (u_i, u_j) \rightarrow \{0, 1\}$ to show whether a certain social link exists or not originally before sampling. For example, if link (u_i, u_j) is a social link in the target network originally before sampling, then $\mathbb{I}(u_i, u_j) = 1$, otherwise it will be equal to 0.

By considering these two terms simultaneously, we could have the diversity of information in the sampled old users' sub-network to be the average diversities of these two parts:

$$\begin{aligned} D(\bar{G}^t_{old}) &= \frac{1}{2}(D_{social}(\bar{G}^t_{old}) + D_{aux}(\bar{G}^t_{old})) \\ &= \frac{1}{2}(\sum_{i=1}^{|\mathcal{U}^t_{old}|} \sum_{j=1}^{|\mathcal{U}^t_{old}|} \frac{1}{\left|\mathcal{E}^t_{old,s}\right|} \cdot \delta_i \cdot \delta_j \times \mathbb{I}(u_i, u_j) + \sum_{i=1}^{|\mathcal{U}^t_{old}|} \frac{1}{\left|\mathcal{U}^t_{old}\right|} \cdot \delta_i^2) \\ &= \boldsymbol{\delta}^\top \cdot (\frac{1}{2\left|\mathcal{E}^t_{old,s}\right|} \cdot \mathbf{A}^t_{old} + \frac{1}{2\left|\mathcal{U}^t_{old}\right|} \cdot \mathbf{I}_{|\mathcal{U}^t_{old}|}) \cdot \boldsymbol{\delta} \end{aligned} \tag{7.23}$$

where matrix $\mathbf{I}_{|\mathcal{U}^t_{old}|}$ is the diagonal identity matrix of dimensions $\left|\mathcal{U}^t_{old}\right| \times \left|\mathcal{U}^t_{old}\right|$ and $\mathbf{A}^t_{old}$ is the adjacency matrix of old users' sub-network.

To ensure that the structure of the original old users' subnetwork is not destroyed, we need to ensure that users with few links could also preserve their links. So, we could add a regularization term to increase the sampling rate for these users as well as their neighbors by maximizing the following terms:

$$Reg(\bar{G}^t_{old}) = \min\{|\Gamma(u_i)|, \min_{u_j \in \Gamma(u_i)} \{|\Gamma(u_j)|\}\} \times \delta_i^2 = \boldsymbol{\delta}^\top \cdot \mathbf{M} \cdot \boldsymbol{\delta} \tag{7.24}$$

where matrix $\mathbf{M}$ is a diagonal matrix with $M_{i,i} = \min\{|\Gamma(u_i)|, \min_{u_j \in \Gamma(u_i)}\{|\Gamma(u_j)|\}\}$ on its diagonal, where $\Gamma(u_i)$ denotes the neighbor set of user u_j. So, if a user or his/her neighbors have few links, then this user as well as his/her neighbors should have higher sampling rates so as to preserve the links between them.

Example 7.4 For example, in Fig. 7.6, we have 6 users. To decide the sampling rate of user u^t_1, we need to consider his/her social structure. We find that since u^t_1's neighbor u^t_2 has no other neighbors except u^t_1. To preserve the social link between u^t_1 and u^t_2 we need to increase the sampling rate of u^t_2. However, the existence probability of link (u^t_1, u^t_2) is also decided by the sampling rate of user u^t_1, which also needs to be increased too.

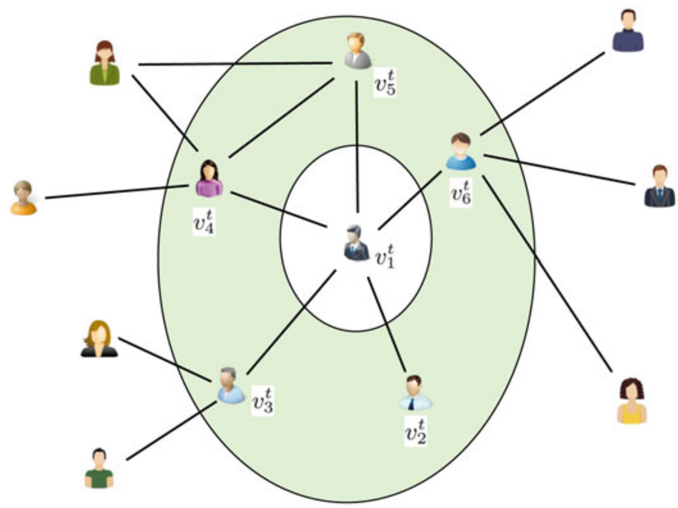


Fig. 7.6 Personalized sampling preserving network structures

Combining the diversity term and the structure preservation term, we could define the regularized diversity of information after sampling to be

$$D_{Reg}(\bar{G}^t_{old}) = D(\bar{G}^t_{old}) + Reg(\bar{G}^t_{old}) = \boldsymbol{\delta}^\top \cdot \mathbf{N} \cdot \boldsymbol{\delta} \tag{7.25}$$

where $\mathbf{N} = \frac{1}{2|\mathcal{U}^t_{old}|} \cdot \mathbf{I}_{|\mathcal{U}^t_{old}|} + \frac{1}{2\left|\mathcal{E}^t_{old,s}\right|} \cdot \mathbf{A}^t_{old} + \mathbf{M}$.

The optimal value of $\boldsymbol{\delta}$ should be able to maximize the relevance of new users' sub-network and old users' as well as the regularized diversity of old users' information in the target network

$$\begin{aligned}
\boldsymbol{\delta} &= \arg\max_{\boldsymbol{\delta}} \; R(\bar{G}^t_{old}, G^t_{new}) + \theta \cdot D_{Reg}(\bar{G}^t_{old}) \\
&= \arg\max_{\boldsymbol{\delta}} \; \boldsymbol{\delta}^\top \mathbf{s} + \theta \cdot \boldsymbol{\delta}^\top \cdot \mathbf{N} \cdot \boldsymbol{\delta} \\
s.t. & \sum_{i=1}^{|\mathcal{U}^t_{old}|} \delta_i = 1 \; and \; \delta_i \geq 0,
\end{aligned} \tag{7.26}$$

where parameter θ denotes the weight of the regularization term on information diversity.

7.4.3.2 TRAD

A traditional supervised link prediction method TRAD (Traditional Link Prediction) can be applied for our task by using the existing links in the target network to train a classifier and applying it to classify the potential social links for new users. In method TRAD, only the target network is used, which consists of new users and unsampled old users. To overcome the differences in information distribution between new users and old users in the target network, we revise it a little bit and get method: TRAD-PS (Traditional Link Prediction with Personalized Sampling). TRAD-PS consists of two steps: (1) personalized sampling of the old users' sub-network with the previous method; (2) usage of similar techniques as TRAD to predict links based on the sampled network. Theoretically, TRAD and TRAD-PS could work well by using information in the target network. However, considering the fact that it is impossible for new users to possess a large amount of information actually, TRAD and

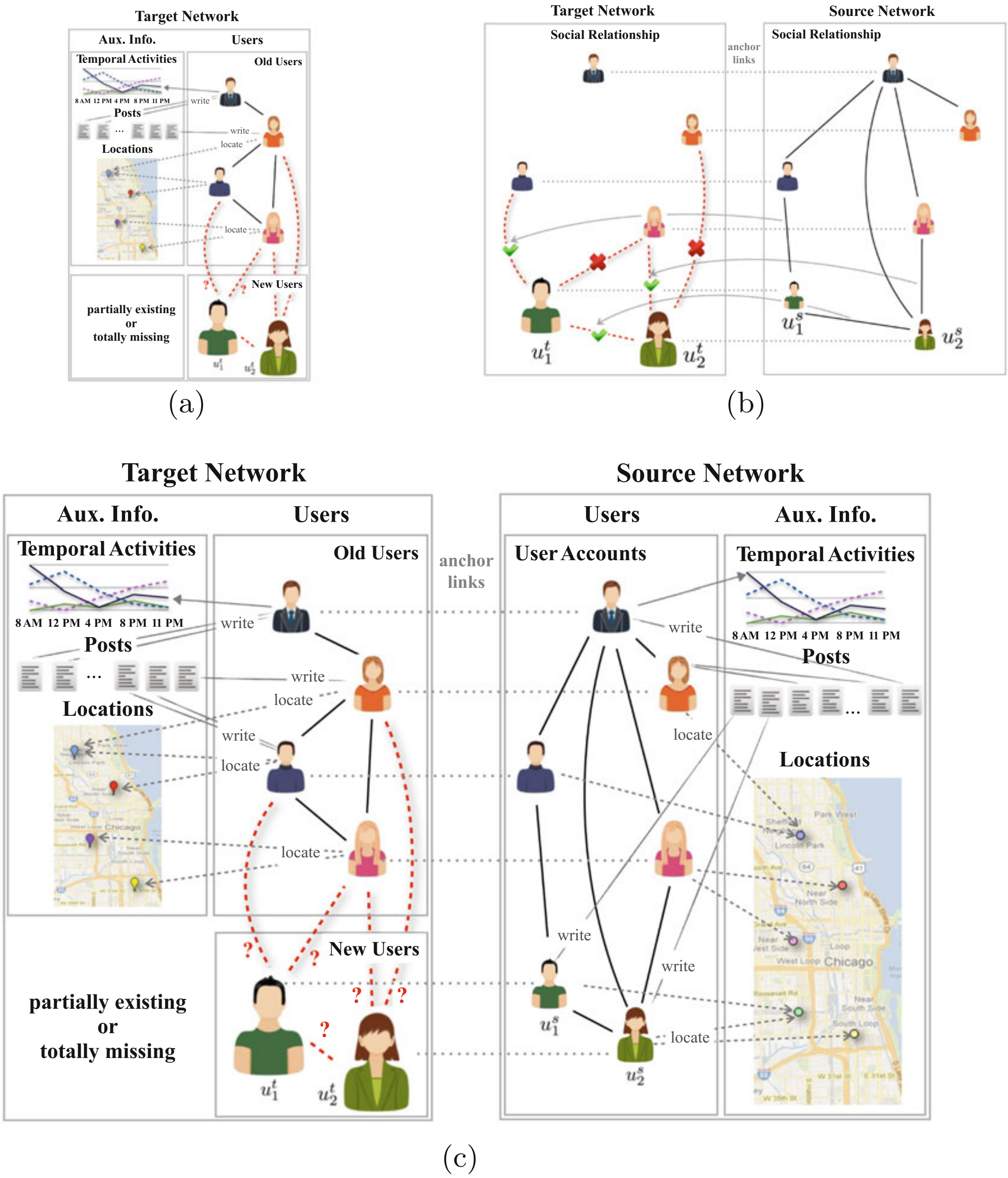


Fig. 7.7 Different methods to predict social link for new users. (**a**) TRAD method. (**b**) NAIVE method. (**c**) SCAN method

TRAD-PS would suffer from the long-standing cold start problem caused by the lack of historical information indicating these new users' preferences. This problem will be even worse when dealing with brand-new users, who have no information at all in the target network.

Example 7.5 For example, in Fig. 7.7a, user u_1^t and u_2^t are two new users in the target network, who possess very few social links with other users and little auxiliary information. We cannot get any information about these two new users and the information we could use is that possessed by other old users. As a result, the links that TRAD and TRAD-PS predicted could hardly be of a high quality.

In order to deal with such a problem, we will introduce a method to use aligned networks simultaneously in the next section.

7.4.4 Cold-Start Link Prediction

In the current problem settings, we have two aligned social networks and the methods proposed in the previous section using the target network may suffer from the cold start problems when processing brand-new users. In this section, we will introduce two methods to utilize the aligned source network to help solve the problem and improve the prediction results.

7.4.4.1 NAIVE

Suppose we have a new user u_i^t in the target network, a naive way to use the aligned source network to recommend social links for user u_i^t is to recommend all the corresponding social links related to this user's aligned account u_i^s in the aligned source network to him/her. Based on this intuition, a cold start link prediction method NAIVE (Naive Link Prediction) as proposed in [57] can be applied. To clarify how NAIVE works in the reality, we will give an example next. And before that, we will introduce a new term *pseudo label* [57] to denote the existence of corresponding links in the aligned source network.

Definition 7.1 (Pseudo Label) The pseudo label of a link (u_i^t, u_j^t) in the target denotes the existence of its corresponding link (u_i^s, u_j^s) in the aligned source network and it is 1 if (u_i^s, u_j^s) exists and 0 otherwise.

Example 7.6 For instance, in Fig. 7.7b, to decide whether to recommend u_1^t to u_2^t in the target network or not, we could find their aligned accounts: u_1^s and u_2^s, and their social link: (u_1^s, u_2^s) in the aligned source network with the help of *anchor links*. We find that u_1^s and u_2^s are friends in the aligned source network and link (u_1^s, u_2^s) exists in the aligned source network. As a result, the pseudo label of link (u_1^t, u_2^t) is 1 and in the target network, we could recommend u_2^t to u_1^t. And that is the reason why the social link between u_1^t and u_2^t is predicted to be existing by method NAIVE. Other links in Fig. 7.7b can be predicted in a similar way.

Method NAIVE is very simple and could work well in addressing the cold start link prediction task even when these new users are brand new, which means that we could overcome the cold start problem by using this method. However, it may still suffer from some disadvantages: (1) the social structures of different networks are not always identical which will degrade the performance of NAIVE a lot; (2) NAIVE only utilizes these new users' social linkage information in the source network and ignores all other information.

7.4.4.2 SCAN

To overcome all these disadvantages mentioned above, a new method SCAN (Supervised Cross Aligned Networks Link Prediction with Personalized Sampling) is proposed in [57]. As shown in Fig. 7.7c, it could use heterogeneous information existing in both the target network and the aligned source and it is built across two aligned social networks. By taking the advantages of the anchor links, we could locate the users' aligned accounts and their information in the aligned source network exactly. If two aligned networks are used simultaneously, different categories of features can be extracted from aligned networks.

To use multiple networks, these feature vectors extracted for the corresponding links in aligned networks are merged into an expanded feature vector. The expanded feature vector together with the labels from the target network are used to build a cross-network classifier to decide the existence of social links related to these new users in the target network. This is how method SCAN works. SCAN is quite stable and could overcome the cold start problem for the reason that the information about all these users in the aligned source network doesn't change much with the variation of the target network and we get the information showing of these new users' preferences from the information he/she leaves in the aligned source network. As the old users' information inside the target network is also used in SCAN, personalized sampling is also conducted to preprocess the old users' information in the target network.

In addition to features mentioned before, SCAN also utilizes the information used by NAIVE, i.e., the *pseudo label* defined before, by treating it as an extra feature.

- **An Extra Feature**: SCAN uses the social link *pseudo label* as an extract feature to denote the existence of the corresponding links in the aligned source network.

Compared with SCAN with NAIVE, SCAN has many advantages: (1) SCAN utilizes multiple categories of information; (2) SCAN can make use of the information hidden in the old users' network by incorporating them into the training set; and (3) SCAN doesn't rely on the assumption that the social relationships in different networks are identical, which is very risky actually.

Compared with TRAD and TRAD-PS, SCAN can solve the cold start problem as it could have access to information owned by these new users in other aligned source networks. Similar to TRAD and TRAD-PS, these new users' information is used if they are not very new and other old users' information in the target is also preprocessed by using the within-network personalized sampling method before the intra-network knowledge transfer.

7.5 Spy Technique Based Inter-Network PU Link Prediction

Besides the link prediction problems in one single target network, some research works have been done on simultaneous link prediction in multiple aligned online social networks concurrently. In the supervised link prediction model introduced before, among all the non-existing social links, a subset of the links can be identified and labeled as the negative instances. However, in the real world, labeling the links which will never be formed can be extremely hard and almost impossible. In this section, we will study the cross-network concurrent link prediction problem with PU learning, and introduce a spy technique based link prediction method MLI proposed in [59].

7.5.1 Cross-Network Concurrent Link Prediction Problem

Traditional link prediction problems which aim at predicting one single kind of links in one network [7,33,45,48] have been studied for many years. Dozens of different link prediction methods have been proposed so far [5,7,25,33,41,45,48]. Conventional link prediction methods usually assume that there exists sufficient information within the network to compute features (e.g., common neighborhoods [13]) for each pair of nodes. However, as proposed in [19, 58], such an assumption can be violated seriously when dealing with social networks containing little information because of the "new network" problems [59].

The *new network problem* can be encountered when online social networks branch into new geographic areas or social groups [58] and information within the new networks can be too sparse to build effective link prediction models. Meanwhile, the recent works [19, 57, 58] notice that users nowadays can participate in multiple online social networks simultaneously. Users who are involved in a new network may use other well-developed networks for a long time, in which they can have plenty of heterogeneous information. To address the new network problem, some papers [57, 58] propose to transfer information from the well-developed networks to overcome the shortage of information problem in the new network. Formally, networks that share some common users are defined as the "*partially aligned networks*" and the common users shared across these *aligned networks* are named as the "*anchor users*" [19,57,58]. Meanwhile, the unshared users are named as the "*non-anchor users*" between the *aligned networks* as introduced in Sect. 3.4.3.

Social networks aligned by the "anchor users" can share common information. Meanwhile, as proposed in [28, 50], different online social networks constructed to provide different services usually have distinct characteristics. Moreover, information in various social networks may be of different distributions [28, 50], which is named as the "*network difference problem*" in [59]. The "*network difference problem*" will be an obstacle in link prediction across *multiple partially aligned networks*, as it is likely that information transferred from other aligned networks could deteriorate the prediction performance in a given network.

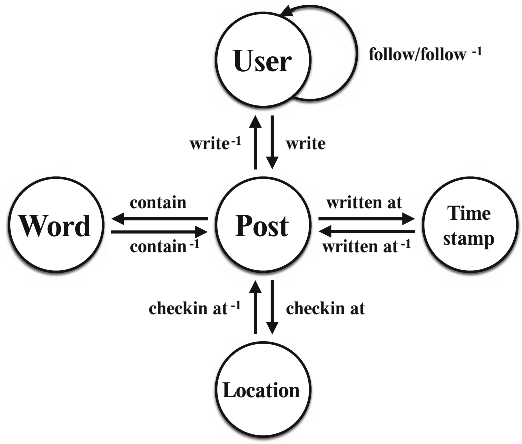
In this section, we want to predict the formation of social links in multiple *partially aligned networks* simultaneously, which is formally defined as the *multi-network link prediction* problem [59]. As introduced at the beginning of this section, the *multi-network link prediction* problem can have very extensive applications in real-world social networks. As a result, the *multi-network link prediction* problem studied in this section is very important for *multiple partially aligned social networks*.

The *multi-network link prediction* problem studied in this section is also very challenging to solve due to: (1) *lack of features*, (2) *network partial alignment problem*, (3) *network difference problem*, and (4) *simultaneous link prediction in multiple networks*. To solve all these above challenges in the *multi-network link prediction* problem, a novel link prediction framework, MLI proposed in [59], will be introduced in this section. Inspired by Sun's work on meta path [40] as a means to capture similarity of nodes, which are not directly connected in heterogeneous information networks, MLI explores the meta path concept to generate useful features. MLI can generate not only intra-network features via "intra-network meta paths," but also inter-network features via "inter-network meta paths" through the *anchor links*. By judiciously selecting the "inter-network meta paths," MLI can take advantage of the commonality among the *multiple partially aligned networks*, while containing the potential negative transfers from network differences. These derived features can greatly improve the effectiveness of MLI in predicting links for each network. Furthermore, MLI is a general link formation prediction framework that solves the *multi-network link prediction* problem and the *link prediction* tasks in different networks can help each other mutually.

7.5.2 Concurrent Link Prediction Problem Formulation

Let $G^{(1)}, G^{(2)}, \ldots, G^{(n)}$ denote n different *heterogeneous online social network*, where the sets of anchor links among them can be represented as $\mathcal{A}^{(1,2)}, \mathcal{A}^{(1,3)}, \ldots, \mathcal{A}^{(n-1,n)}$. The user set and existing social link set of $G^{(i)}$ can be represented as $\mathcal{U}^{(i)}$ and $\mathcal{E}_{u,u}^{(i)}$, respectively. In network $G^{(i)}$, all the existing links are the formed links and, as a result, the formed links of $G^{(i)}$ can be represented as $\mathcal{P}^{(i)}$, where $\mathcal{P}^{(i)} = \mathcal{E}_{u,u}^{(i)}$. Furthermore, a large set of unconnected user pairs are referred to as the unconnected

Fig. 7.8 Schema of heterogeneous network



links, $\bar{\mathcal{U}}^{(i)}$, and can be extracted from network $G^{(i)}$: $\bar{\mathcal{U}}^{(i)} = \mathcal{U}^{(i)} \times \mathcal{U}^{(i)} \setminus \mathcal{P}^{(i)}$. However, no information about links that will never be formed can be obtained from the network. With the formed link set $\mathcal{P}^{(i)}$ and unconnected link set $\bar{\mathcal{U}}^{(i)}$, the *link formation prediction* problem can be formulated as a *PU link prediction* problem.

Formally, let $\{\mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(n)}\}$, $\{\bar{\mathcal{U}}^{(1)}, \ldots, \bar{\mathcal{U}}^{(n)}\}$ and $\{\mathcal{L}^{(1)}, \ldots, \mathcal{L}^{(n)}\}$ be the sets of formed links, unconnected links, and links to be predicted of $G^{(1)}, G^{(2)}, \ldots, G^{(n)}$, respectively. With the formed and unconnected links of $G^{(1)}, G^{(2)}, \ldots, G^{(n)}$, we can solve the *multi-network link prediction* problem as the *concurrent PU link prediction* problem.

In the following subsections, we will introduce MLI to solve the *multi-network link prediction* problem. This section includes 3 parts: (1) social meta path based feature extraction and selection; (2) PU link prediction; (3) multi-network concurrent link prediction framework.

7.5.3 Social Meta Path Definition and Selection

Before talking about the link prediction methods, we will introduce the features extracted from the *partially aligned networks* in this subsection at first. The feature extraction in MLI is based on the meta paths as defined in Sect. 3.5. Based on the schema of the network studied in this section, shown in Fig. 7.8, we can define many different kinds of *homogeneous and heterogeneous intra-network social meta paths* for the network, whose physical meanings and notations are listed as follows:

Homogeneous Intra-Network Social Meta Path

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

Heterogeneous Intra-Network 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 Check-ins*: 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 Features: These meta paths can actually cover a large number of path instances connecting users in the network. Formally, we denote that node n (or link l) is an instance of node type T (or link type R) in the network as $n \in T$ (or $l \in R$). Identity function $\mathbb{I}(a, A) = \begin{cases} 1, & \text{if } a \in A \\ 0, & otherwise, \end{cases}$ can check whether node/link a is an instance of node/link type A in the network. To consider the effect of the unconnected links when extracting features for social links in the network, the *Intra-Network Social Meta Path based Features* can be formally defined as follows:

Definition 7.2 (Intra-Network 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 path instances between u and v in the network:

$$x(u, v) = \mathbb{I}(u, T_1)\mathbb{I}(v, T_k) \sum_{n_1 \in \{u\}, n_2 \in T_2, \ldots, n_k \in \{v\}} \prod_{i=1}^{k-1} p(n_i, n_{i+1})\mathbb{I}((n_i, n_{i+1}), R_i), \tag{7.27}$$

where $p(n_i, n_{i+1}) = 1.0$ if $(n_i, n_{i+1}) \in \mathcal{E}_{u,u}$ and otherwise, $p(n_i, n_{i+1})$ denotes the *formation probability* of link (n_i, n_{i+1}) to be introduced in Sect. 7.5.4.

Features extracted by MLI based on $\Phi = \{\Phi_1, \ldots, \Phi_6\}$ are named as the *intra-network social meta path* based social features. (Φ_0 will be used in the following subsection only.)
Inter-Network Social Meta Paths: When a network is very new, features extracted based on *intra-network social meta paths* can be very sparse, as there exist few connections in the network.

Example 7.7 Consider, for example, in Fig. 7.9, we want to predict whether social link $(A^{(1)}, B^{(1)})$ in network $G^{(1)}$ will be formed or not. Merely based on the *intra-network social meta paths*, the feature vector of extracted for link $(A^{(1)}, B^{(1)})$ will be $\mathbf{0}$. However, we find that $A^{(1)}$ and $B^{(1)}$ can be correlated actually with various inter-network paths, e.g., $B^{(1)} \to B^{(2)} \to A^{(2)} \to A^{(1)}$, $B^{(1)} \to B^{(2)} \to F^{(2)} \to A^{(2)} \to A^{(1)}$ and $B^{(1)} \to B^{(2)} \to G^{(2)} \to A^{(2)} \to A^{(1)}$.

By following this idea, MLI proposes to transfer useful information from aligned networks with the following *anchor meta path* and the *inter-network social meta paths*, whose formal definitions are available in Sect. 3.5. In MLI, we are mainly concerned about *inter-network meta path* starting and ending with users, which are named as the *inter-network social meta path*. Let $\Upsilon(U^{(i)}, U^{(j)})$ denote the *anchor meta path* defined between networks $G^{(i)}$ and $G^{(j)}$. The 4 specific *inter-network social meta paths* used in MLI include:

- **Category 1**: $\Upsilon(U^{(i)}, U^{(j)}) \circ (\Phi(U^{(j)}, U^{(j)}) \cup \Phi_0(U^{(j)}, U^{(j)})) \circ \Upsilon(U^{(j)}, U^{(i)})$, whose notation is $\Psi_1(U^{(i)}, U^{(i)})$;
- **Category 2.**: $(\Phi(U^{(i)}, U^{(i)}) \cup \Phi_0(U^{(i)}, U^{(i)})) \circ \Upsilon(U^{(i)}, U^{(j)}) \circ (\Phi(U^{(j)}, U^{(j)}) \cup \Phi_0(U^{(j)}, U^{(j)})) \circ \Upsilon(U^{(j)}, U^{(i)})$, whose notation is $\Psi_2(U^{(i)}, U^{(i)})$;
- **Category 3.**: $\Upsilon(U^{(i)}, U^{(j)}) \circ (\Phi(U^{(j)}, U^{(j)}) \cup \Phi_0(U^{(j)}, U^{(j)})) \circ \Upsilon(U^{(j)}, U^{(i)}) \circ (\Phi(U^{(i)}, U^{(i)}) \cup \Phi_0(U^{(i)}, U^{(i)}))$, whose notation is $\Psi_3(U^{(i)}, U^{(i)})$;
- **Category 4.**: $(\Phi(U^{(i)}, U^{(i)}) \cup \Phi_0(U^{(i)}, U^{(i)})) \circ \Upsilon(U^{(i)}, U^{(j)}) \circ (\Phi(U^{(j)}, U^{(j)}) \cup \Phi_0(U^{(j)}, U^{(j)})) \circ \Upsilon(U^{(j)}, U^{(i)}) \circ (\Phi(U^{(i)}, U^{(i)}) \cup \Phi_0(U^{(i)}, U^{(i)}))$, whose notation is $\Psi_4(U^{(i)}, U^{(i)})$;

where $\Phi(U^{(i)}, U^{(i)}) \cup \Phi_0(U^{(i)}, U^{(i)}) = \{\Phi_0(U^{(i)}, U^{(i)}), \ldots, \Phi_6(U^{(i)}, U^{(i)})\}$ denote the 7 *intra-network social meta paths* in network $G^{(i)}$ introduced before.

Let $\Psi = \{\Psi_1, \Psi_2, \Psi_3, \Psi_4\}$. Ψ is a comprehensive *inter-network social meta path* set and features extracted based on Ψ can transfer information for both anchor users and non-anchor users from other aligned networks.

Example 7.8 For example, in Fig. 7.9, by following path "$B^{(1)} \rightarrow B^{(2)} \rightarrow A^{(2)} \rightarrow A^{(1)}$," we can go from an *anchor user* $B^{(1)}$ to another *anchor user* $A^{(1)}$ and such path is an instance of $\Psi_1(U^{(1)}, U^{(1)})$; by following path $C^{(1)} \rightarrow A^{(1)} \rightarrow A^{(2)} \rightarrow D^{(2)} \rightarrow D^{(1)}$, we can go from a *non-anchor user* $C^{(1)}$ to an *anchor user* $D^{(1)}$, which is an instance of $\Psi_2(U^{(1)}, U^{(1)})$; in addition, by following path $C^{(1)} \rightarrow A^{(1)} \rightarrow A^{(2)} \rightarrow B^{(2)} \rightarrow B^{(1)} \rightarrow E^{(1)}$, we can go from a *non-anchor user* $C^{(1)}$ to another *non-anchor user* $E^{(1)}$, which is an instance of $\Psi_4(U^{(1)}, U^{(1)})$.

Social Meta Path Selection: As introduced in Sect. 7.5.1, information transferred from aligned networks is helpful for improving link prediction performance in a given network but can be misleading as well, which is called the *network difference problem*. To solve the *network difference*

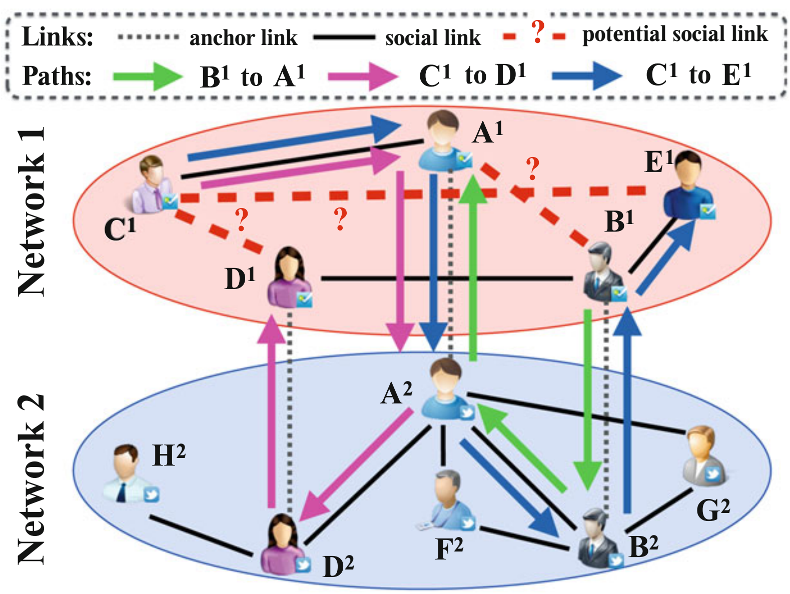


Fig. 7.9 Meta path across aligned networks

problem, MLI proposes to rank and select the top K features from the feature vector extracted based on the *intra-network* and *inter-network social meta paths*, $[\mathbf{x}_\Phi^\top, \mathbf{x}_\Psi^\top]^\top$, from the multiple *partially aligned heterogeneous networks*.

Let variable $X_i \in [\mathbf{x}_\Phi^\top, \mathbf{x}_\Psi^\top]^\top$ be a feature extracted based on a meta path in $\{\Phi, \Psi\}$ and variable Y be the *label*. $P(Y = y)$ denotes the *prior probability* that links in the training set having label y and $P(X_i = x)$ represents the *frequency* that feature X_i has value x. Information theory related measure *mutual information* (mi) [43] is used as the ranking criteria:

$$mi(X_i) = \sum_x \sum_y P(X_i = x, Y = y) \log \frac{P(X_i = x, Y = y)}{P(X_i = x)P(Y = y)} \tag{7.28}$$

Let $[\bar{\mathbf{x}}_\Phi^\top, \bar{\mathbf{x}}_\Psi^\top]^\top$ be the features of the top K *mi* score selected from $[\mathbf{x}_\Phi^\top, \mathbf{x}_\Psi^\top]^\top$. In the next subsection, we will use the selected feature vector $[\bar{\mathbf{x}}_\Phi^\top, \bar{\mathbf{x}}_\Psi^\top]^\top$ to build a novel PU link prediction model.

7.5.4 Spy Technique Based PU Link Prediction

In this subsection, we will first introduce a method to solve the *PU link prediction* problem in one single network. As introduced in Sect. 7.5.2, from a given network, e.g., G, we can get two disjoint sets of links: connected (i.e., formed) links $\mathcal{P}$ and unconnected links $\mathcal{U}$. To differentiate these links, we define a new concept "*connection state*," z, to show whether a link is connected (i.e., formed) or unconnected in network G. For a given link l, if l is connected in the network, then $z(l) = +1$; otherwise, $z(l) = -1$. As a result, we can have the "*connection states*" of links in $\mathcal{P}$ and $\mathcal{U}$ to be: $z(\mathcal{P}) = \mathbf{+1}$ and $z(\mathcal{U}) = \mathbf{-1}$.

Besides the "*connection state*," links in the network can also have their own "*labels*," y, which can represent whether a link is to be formed or will never be formed in the network. For a given link l, if l has been formed or to be formed, then $y(l) = +1$; otherwise, $y(l) = -1$. Similarly, we can have the "*labels*" of links in $\mathcal{P}$ and $\mathcal{U}$ to be: $y(\mathcal{P}) = \mathbf{+1}$ but $y(\mathcal{U})$ can be either $+1$ or -1, as $\mathcal{U}$ can contain both links to be formed and links that will never be formed.

By using $\mathcal{P}$ and $\mathcal{U}$ as the positive and negative training sets, we can build a *link connection prediction model* $\mathcal{M}_c$, which can be applied to predict whether a link exists in the original network, i.e., the *connection state* of a link. Let l be a link to be predicted, by applying $\mathcal{M}_c$ to classify l, we can get the *connection probability* of l to be:

Definition 7.3 (Connection Probability) The probability that link l's *connection states* is predicted to be *connected* (i.e., $z(l) = +1$) is formally defined as the *connection probability* of link l: $p(z(l) = +1|\mathbf{x}(l))$, where $\mathbf{x}(l) = [\bar{\mathbf{x}}_\Phi(l)^\top, \bar{\mathbf{x}}_\Psi(l)^\top]^\top$.

Meanwhile, if we can obtain a set of links that "will never be formed," i.e., "-1" links, from the network, which together with $\mathcal{P}$ ("+1" links) can be used to build a *link formation prediction model* $\mathcal{M}_f$. Here, model $\mathcal{M}_f$ can be used to get the *formation probability* of l to be:

Definition 7.4 (Formation Probability) The probability that link l's *label* is predicted to be *formed or will be formed* (i.e., $y(l) = +1$) is formally defined as the *formation probability* of link l: $p(y(l) = +1|\mathbf{x}(l))$.

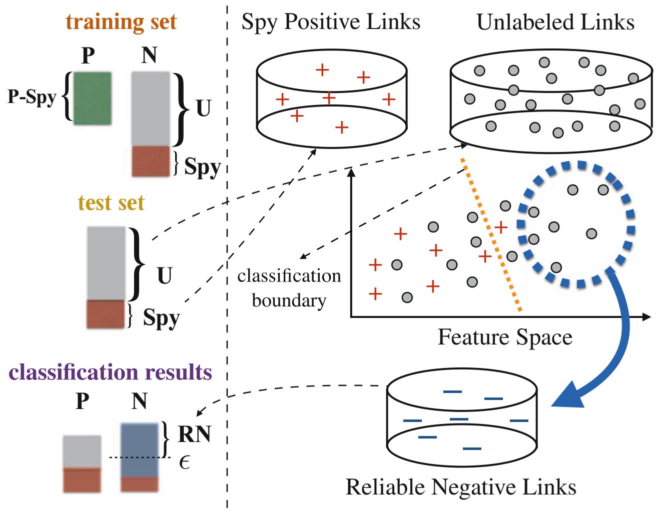


Fig. 7.10 PU link prediction

However, from the network, we have no information about "links that will never be formed" (i.e., "-1" links). As a result, the *formation probabilities* of potential links that we aim to obtain as proposed in Sect. 7.5.2 can be very challenging to calculate. Meanwhile, the correlation between link l's *connection probability* and *formation probability* has been proved in existing works [11] to be:

$$p(y(l) = +1|\mathbf{x}(l)) \propto p(z(l) = +1|\mathbf{x}(l)). \tag{7.29}$$

In other words, for links whose *connection probabilities* are low, their *formation probabilities* will be relatively low as well. This rule can be utilized to extract links which can be more likely to be the reliable "-1" links from the network. The *link connection prediction model* $\mathcal{M}_c$ built with $\mathcal{P}$ and $\bar{\mathcal{U}}$ can be applied to classify links in $\bar{\mathcal{U}}$ to extract the *reliable negative link set*.

Definition 7.5 (Reliable Negative Link Set) The *reliable negative links* in the *unconnected link* set $\bar{\mathcal{U}}$ are those whose *connection probabilities* predicted by the *link connection prediction model*, $\mathcal{M}_c$, are lower than threshold $\epsilon \in [0, 1]$:

$$\mathcal{RN} = \{l | l \in \bar{\mathcal{U}}, p(z(l) = +1|\mathbf{x}(l)) < \epsilon\}. \tag{7.30}$$

Some heuristic based methods have been proposed to set the optimal threshold ϵ, e.g., the *spy technique* proposed in [24]. As shown in Fig. 7.10, we randomly selected a subset of links in $\mathcal{P}$ as the spy, $\mathcal{SP}$, whose proportion is controlled by $s\%$ ($s\% = 15\%$ is used as the default sample rate as introduced in [59]). Sets $(\mathcal{P} \setminus \mathcal{SP})$ and $(\bar{\mathcal{U}} \cup \mathcal{SP})$ are used as positive and negative training sets to the *spy prediction* model, $\mathcal{M}_s$. By applying $\mathcal{M}_s$ to classify links in $(\bar{\mathcal{U}} \cup \mathcal{SP})$, we can get their *connection probabilities* to be:

$$p(z(l) = +1|\mathbf{x}(l)), l \in (\bar{\mathcal{U}} \cup \mathcal{SP}), \tag{7.31}$$

and parameter ϵ is set as the minimal *connection probability* of spy links in $\mathcal{SP}$:

$$\epsilon = \min_{l \in \mathcal{SP}} p(z(l) = +1|\mathbf{x}(l)). \tag{7.32}$$

With the extracted *reliable negative link set* $\mathcal{RN}$, we can solve the *PU link prediction* problem with *classification based link prediction methods*, where $\mathcal{P}$ and $\mathcal{RN}$ are used as the positive and negative training sets, respectively. Meanwhile, when applying the built model to predict links in $\mathcal{L}^{(i)}$, their optimal labels, i.e., $\hat{\mathcal{Y}}^{(i)}$, should be those which can maximize the following *formation probabilities*:

$$\begin{aligned}\hat{\mathcal{Y}}^{(i)} &= \arg\max_{\mathcal{Y}^{(i)}} p(y(\mathcal{L}^{(i)}) = \mathcal{Y}^{(i)} | G^{(1)}, G^{(2)}, \ldots, G^{(k)}) \\ &= \arg\max_{\mathcal{Y}^{(i)}} p(y(\mathcal{L}^{(i)}) = \mathcal{Y}^{(i)} | \left[\bar{\mathbf{x}}_{\Phi}(\mathcal{L}^{(i)})^\top, \bar{\mathbf{x}}_{\Psi}(\mathcal{L}^{(i)})^\top\right]^\top) \end{aligned} \tag{7.33}$$

where $y(\mathcal{L}^{(i)}) = \mathcal{Y}^{(i)}$ represents that links in $\mathcal{L}^{(i)}$ have labels $\mathcal{Y}^{(i)}$.

7.5.5 Multi-Network Concurrent PU Link Prediction Framework

Method MLI to be introduced in this part is a general link prediction framework and can be applied to predict social links in *n partially aligned networks* simultaneously. When it comes to n partially aligned network formulated in Sect. 7.5.2, the optimal labels of potential links $\{\mathcal{L}^{(1)}, \mathcal{L}^{(2)}, \ldots, \mathcal{L}^{(n)}\}$ of networks $G^{(1)}, G^{(2)}, \ldots, G^{(n)}$ will be:

$$\begin{aligned}\hat{\mathcal{Y}}^{(1)}, \hat{\mathcal{Y}}^{(2)}, \ldots, \hat{\mathcal{Y}}^{(n)} = \arg \max_{\mathcal{Y}^{(1)},\mathcal{Y}^{(2)},\ldots,\mathcal{Y}^{(n)}} & p(y(\mathcal{L}^{(1)}) = \mathcal{Y}^{(1)}, y(\mathcal{L}^{(2)}) = \mathcal{Y}^{(2)}, \\ & \ldots, y(\mathcal{L}^{(n)}) = \mathcal{Y}^{(n)} | G^{(1)}, G^{(2)}, \ldots, G^{(n)}) \end{aligned} \tag{7.34}$$

The above target function is very complex to solve and, in [59], MLI obtains the solution by updating one variable, e.g., $\mathcal{Y}^{(1)}$, and fix other variables, e.g., $\mathcal{Y}^{(2)}, \ldots, \mathcal{Y}^{(n)}$, alternatively with the following equation:

$$\begin{cases} (\hat{\mathcal{Y}}^{(1)})^{(\tau)} & = \arg\max_{\mathcal{Y}^{(1)}} p(y(\mathcal{L}^{(1)}) = \mathcal{Y}^{(1)} | G^{(1)}, G^{(2)}, \ldots, G^{(n)}, \\ & \qquad (\hat{\mathcal{Y}}^{(2)})^{(\tau-1)}, (\hat{\mathcal{Y}}^{(3)})^{(\tau-1)}, \ldots, (\hat{\mathcal{Y}}^{(n)})^{(\tau-1)}) \\ (\hat{\mathcal{Y}}^{(2)})^{(\tau)} & = \arg\max_{\mathcal{Y}^{(2)}} p(y(\mathcal{L}^{(2)}) = \mathcal{Y}^{(2)} | G^{(1)}, G^{(2)}, \ldots, G^{(n)}, \\ & \qquad (\hat{\mathcal{Y}}^{(1)})^{(\tau)}, (\hat{\mathcal{Y}}^{(3)})^{(\tau-1)}, \ldots, (\hat{\mathcal{Y}}^{(n)})^{(\tau-1)}) \\ & \ldots\ldots \\ (\hat{\mathcal{Y}}^{(n)})^{(\tau)} & = \arg\max_{\mathcal{Y}^{(n)}} p(y(\mathcal{L}^{(n)}) = \mathcal{Y}^{(n)} | G^{(1)}, G^{(2)}, \ldots, G^{(n)}, \\ & \qquad (\hat{\mathcal{Y}}^{(1)})^{(\tau)}, (\hat{\mathcal{Y}}^{(2)})^{(\tau)}, \ldots, (\hat{\mathcal{Y}}^{(n-1)})^{(\tau)}) \end{cases} \tag{7.35}$$

The architecture of framework MLI is shown in Fig. 7.11. When predicting social links in network $G^{(i)}$, MLI can extract features based on the *intra-network social meta path*, $\mathbf{x}_{\Phi}$, extracted from $G^{(i)}$ and those extracted based on the *inter-network social meta path*, $\mathbf{x}_{\Psi}$, across $G^{(1)}, G^{(2)}, \ldots, G^{(i-1)}, G^{(i+1)}, \ldots, G^{(n)}$ for links in $\mathcal{P}^{(i)}, \bar{\mathcal{U}}^{(i)}$ and $\mathcal{L}^{(i)}$. Feature vectors $\mathbf{x}_{\Phi}(\mathcal{P}), \mathbf{x}_{\Phi}(\bar{\mathcal{U}})$ and $\mathbf{x}_{\Psi}(\mathcal{P}), \mathbf{x}_{\Psi}(\bar{\mathcal{U}})$ as well as the labels, $y(\mathcal{P})$, $y(\bar{\mathcal{U}})$, of links in $\mathcal{P}$ and $\bar{\mathcal{U}}$ are passed to the PU link prediction model $\mathcal{M}^{(i)}$ and the meta path selection model $\mathcal{MS}^{(i)}$. The formation probabilities of links in $\mathcal{L}^{(i)}$ predicted by model $\mathcal{M}^{(i)}$ will be used to update the network by replacing the weights of $\mathcal{L}^{(i)}$ with the newly predicted formation probabilities. The initial weights of these potential links in $\mathcal{L}^{(i)}$ are set as 0 (i.e., the *formation probability* of links mentioned in Definition 7.2). After finishing these steps on $G^{(i)}$, we

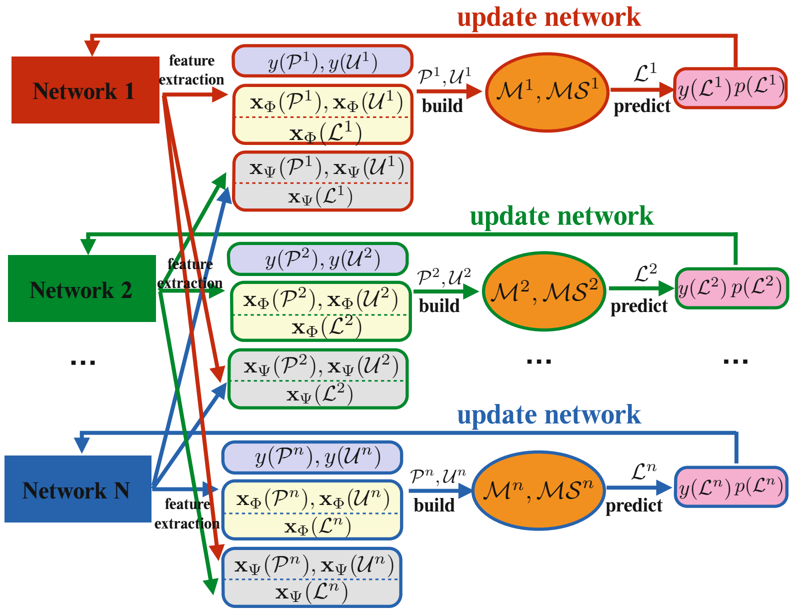


Fig. 7.11 Multi-PU link prediction framework

will move to conduct similar operations on $G^{(i+1)}$. We predict links in $G^{(1)}$ to $G^{(n)}$ alternatively in a sequence until the results in all of these networks converge.

7.6 Sparse and Low Rank Matrix Estimation Based PU Link Prediction

Different online social networks usually have different functions, and information in them follows totally different distributions. When predicting the links across multiple aligned online social networks, the link prediction models aforementioned, which merely append the feature vectors from different sources, can hardly address the domain difference problem at all. In this section, we will introduce a new cross-network link prediction model proposed in [61], which embeds the feature vectors of links from aligned networks into a shared feature space. The knowledge from the source networks is transferred to the target network in the shared feature space.

7.6.1 Problem Description

In this section, we will study the link prediction problem for the target network, which is aligned with multiple source networks concurrently. Formally, the problem is named as the "Social Link Transfer" (SLT) problem. Formally, given the *multiple aligned online social networks* $\mathcal{G} = (\{G^t, G^{(1)}, G^{(2)}, \ldots, G^{(K)}\}, \{\mathcal{A}^{(t,1)}, \mathcal{A}^{(t,2)}, \ldots, \mathcal{A}^{(K-1,K)}\})$, the SLT problem to be studied in this section aims at inferring the potential social connections among users in the target network G^t with information across all these networks. Formally, based on information available in $\mathcal{G}$, the objective of SLT is to build a social link prediction function $S : \mathcal{U}^t \times \mathcal{U}^t \setminus \mathcal{E}_u^t \to [0, 1]$ to infer the confidence scores of all the potential social connections among the users in the target network G^t, where $\mathcal{U}^t$ and $\mathcal{E}_u^t$ represent the existing users and social links in G^t, respectively.

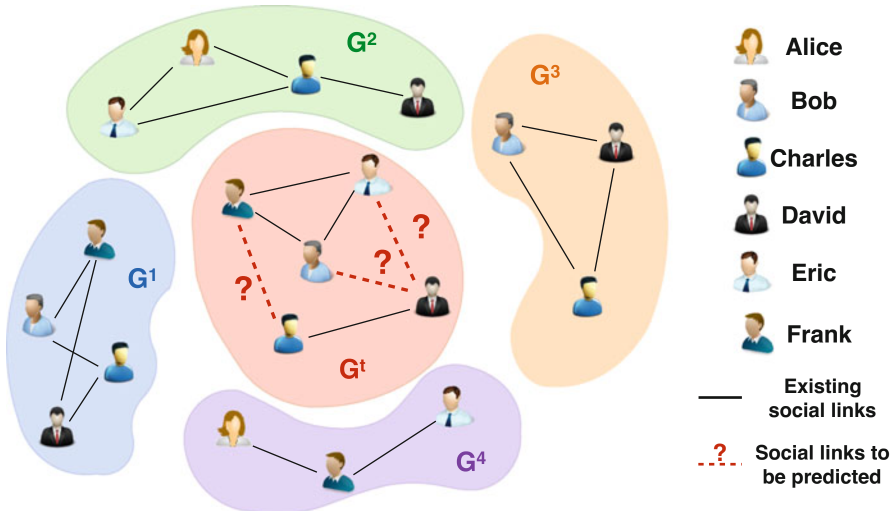


Fig. 7.12 An example of link prediction across aligned networks

Example 7.9 An example to illustrate the SLT problem is provided in Fig. 7.12. In Fig. 7.12, network G^t is the target network, and $G^{(1)}, \ldots, G^{(4)}$ are the other aligned source networks, which share a number of common users with G^t. With the information across networks $\{G^t, G^{(1)}, \ldots, G^{(4)}\}$, the objective of SLT is to infer potential social links (i.e., the red dashed lines) to be formed in the target network G^t.

The SLT problem studied in this section is based on the same setting as those in [57, 59], but we will introduce a new model to address the problem. We summarize the differences of this work from these existing works as follows. Firstly, the link prediction model proposed in this section is based on the matrix estimation, which is totally different from the classification based models proposed in [57, 59] and will not suffer from the class imbalance problem. Secondly, considering the connections among users in the networks are usually very sparse and users tend to form densely connected local communities, a sparse regularizer and a low-rank regularizer are incorporated in the objective function. Thirdly, these existing works [57, 59] transfer information across different networks without considering the domain differences. Meanwhile, based on the known anchor and social link information, our model overcomes the domain difference problem by mapping the feature vectors extracted for links from the aligned networks to a shared lower-dimensional latent feature space instead.

The SLT problem studied in this section is very hard to solve mainly due to the following challenges caused by (1) the *heterogeneity of networks*, (2) the *multiple aligned networks* setting, (3) the *sparse and low-rank property* of the target network, and (4) the *objective function* is hard to solve. To overcome these challenges, a novel link prediction model named SLAMPRED (*S*parse *L*ow-r*A*nk *M*atrix estimation based *Pred*iction) [61] will be introduced in this part. SLAMPRED formulates the link prediction problem as a sparse and low-rank matrix estimation problem. Heterogeneous information is used to calculate the similarity among users, and similar users tend be linked. With the existing anchor and social link information, SLAMPRED proposes to map the feature vectors of the social links extracted from the target and other aligned source networks to a common low-dimensional latent feature space. Two regularizers are introduced in the objective function of SLAMPRED to preserve the sparse and low-rank properties. Furthermore, SLAMPRED solves the objective function with the iterative CCCP (convex concave procedure), and in each iteration the involved non-differentiable sparsity and low-rank regularizers are effectively handled by the proximal operators.

In the following parts, we will first introduce the link prediction model built with the observed network connection information and other heterogeneous attribute information available in the target network. After that, we will talk about the target network link prediction problem with information across multiple aligned networks, where the features extracted from different networks are projected to a lower-dimensional feature space to accommodate the domain differences. Finally, we will introduce the joint optimization objective function, which can be resolved by the proximal operator based iterative CCCP algorithm effectively.

7.6.2 Intra-Network Link Prediction

Users' diverse online social activities may generate heterogeneous information in the online social networks, which include both the network structure information and the different categories of attribute information about the users. In this subsection, we will introduce the link prediction method with the heterogeneous information available in the target network.

7.6.2.1 Intra-Network Link Prediction with Link Information

Given the target network G^t involving users $\mathcal{U}^t$, we can represent the observed social connection among the users with the binary social adjacency matrix $\mathbf{A}^t \in \{0, 1\}^{|\mathcal{U}^t| \times |\mathcal{U}^t|}$, where entry $A^t(i, j) = 1$ iff the corresponding social link (u_i^t, u_j^t) exists between users u_i^t and u_j^t in G^t. In the SLT problem, our objective is to infer the potential unobserved social links for the target network, which can be achieved by finding a sparse and low-rank predictor matrix $\mathbf{S} \in \mathcal{S}$ from some convex admissible set $\mathcal{S} \subset \mathbb{R}^{|\mathcal{U}^t| \times |\mathcal{U}^t|}$. Meanwhile, the inconsistency between the inferred matrix $\mathbf{S}$ and the observed social adjacency matrix $\mathbf{A}^t$ can be represented as the loss function $l(\mathbf{S}, \mathbf{A}^t)$. The optimal social link predictor for the target network can be achieved by minimizing the loss term, i.e.,

$$\arg\min_{\mathbf{S} \in \mathcal{S}} l(\mathbf{S}, \mathbf{A}^t). \tag{7.36}$$

The loss function $l(\mathbf{S}, \mathbf{A}^t)$ can be defined in many different ways, and the *loss function* can be approximated by counting the loss introduced by the existing social links in $\mathcal{E}_u^t$, i.e.,

$$l(\mathbf{S}, \mathbf{A}^t) = \frac{1}{|\mathcal{E}_u^t|} \sum_{(u_i^t, u_j^t) \in \mathcal{E}_u^t} \mathbb{1}\left(\left(A^t(i, j) - \frac{1}{2}\right) \cdot S(i, j) \leq 0\right). \tag{7.37}$$

7.6.2.2 Intra-Network Link Prediction with Heterogeneous Attribute Information

Besides the connection information, there also exists a large amount of attribute information available in the target network, e.g., *location check-in records*, *online social activity temporal patterns*, and *text usage patterns*, etc. Based on the attribute information, a set of features can be extracted for all the potential user pairs to denote their closeness, which are called the *intimacy features* formally. For instance, given a user pair (u_i^t, u_j^t) in the target network, we can represent its *intimacy features* as vector $\mathbf{x}_{i,j}^t \in \mathbb{R}^{d^t}$ (d^t denotes the extracted intimacy feature number). According to the existing works [14, 57], different intimacy features can be extracted from the attribute information.

More generally, we can represent the feature vectors extracted for user pairs as a 3-way tensor $\mathbf{X}^t \in \mathbb{R}^{d^t \times |\mathcal{U}^t| \times |\mathcal{U}^t|}$, where slice $\mathbf{X}^t(k, :, :)$ denotes all the k_{th} intimacy features among all the user pairs. In online social networks, *homophily* principle [27] has been observed to widely structure the users' online social connections, and users who are close to each other are more likely to be friends. Based on such an intuition, we can infer the potential social connection matrix $\mathbf{S}$ by maximizing the overall intimacy scores of the inferred new social connections, i.e.,

$$\arg \max_{\mathbf{S} \in \mathcal{S}} int(\mathbf{S}, \mathbf{X}^t). \tag{7.38}$$

SLAMPRED proposes to define the intimacy score term $int(\mathbf{S}, \mathbf{X}^t)$ by enumerating and summing the *intimacy scores* of the inferred social connections, i.e.,

$$int(\mathbf{S}, \mathbf{X}^t) = \sum_{k=1}^{d^t} \left\| \mathbf{S} \circ \mathbf{X}^t(k, :, :) \right\|_1, \tag{7.39}$$

where operator $\circ$ denotes the Hadamard product (i.e., entrywise product) of two matrices.

7.6.2.3 Joint Optimization Function for Intra-Network Link Prediction

By considering the link and attribute information in the target network at the same time, we can represent the joint optimization for link prediction in the target network to be

$$\arg \min_{\mathbf{S} \in \mathcal{S}} l(\mathbf{S}, \mathbf{A}^t) - \alpha^t \cdot int(\mathbf{S}, \mathbf{X}^t) + \gamma \cdot \|\mathbf{S}\|_1 + \tau \cdot \|\mathbf{S}\|_* . \tag{7.40}$$

Considering that the social connections in online social networks are usually very sparse and of low-rank, the regularizers $\|\mathbf{S}\|_1$ and $\|\mathbf{S}\|_*$ are added to preserve the *sparse* and *low rank* properties of the inferred predictor matrix $\mathbf{S}$. Parameters α^t, γ, τ denote the importance scalars of different terms in the objective function.

7.6.3 Inter-Network Link Prediction

Besides the information available in the target network, a large amount of information about the users' social activities is available in other external source networks as well, which can be transferred to the target network to help improve the link prediction results, especially when the target network suffers from information sparsity problem. To be general, we can represent the *intimacy* features extracted for user pairs in source network $G^{(i)}$ ($i \in \{1, 2, \ldots, K\}$) as a 3-way tensor $\mathbf{X}^{(i)} \in \mathbb{R}^{d^{(i)} \times |\mathcal{U}^{(i)}| \times |\mathcal{U}^{(i)}|}$, where $\mathcal{U}^{(i)}$ denotes the user set in $G^{(i)}$ and $d^{(i)}$ is the extracted feature number.

Meanwhile, different online social networks are constructed for different purposes, information from which may follow totally different distributions actually. To adopt the information domains of these different aligned networks, SLAMPRED proposes to project the extracted feature vectors from different networks (both G^t and aligned source networks $G^{(1)}, \ldots, G^{(K)}$) to a common lower-dimensional feature space instead. Given the $K+1$ partially aligned social networks, we formulate the information domain adaption problem as a mapping function inference problem instead. Our objective is to construct $K+1$ mapping functions, $f^t : \mathbb{R}^{d^t} \to \mathbb{R}^c, \ldots, f^{(K)} : \mathbb{R}^{d^{(K)}} \to \mathbb{R}^c$ to map the $K+1$ input features to a new c-dimensional latent space, where certain properties about the networks are still preserved.

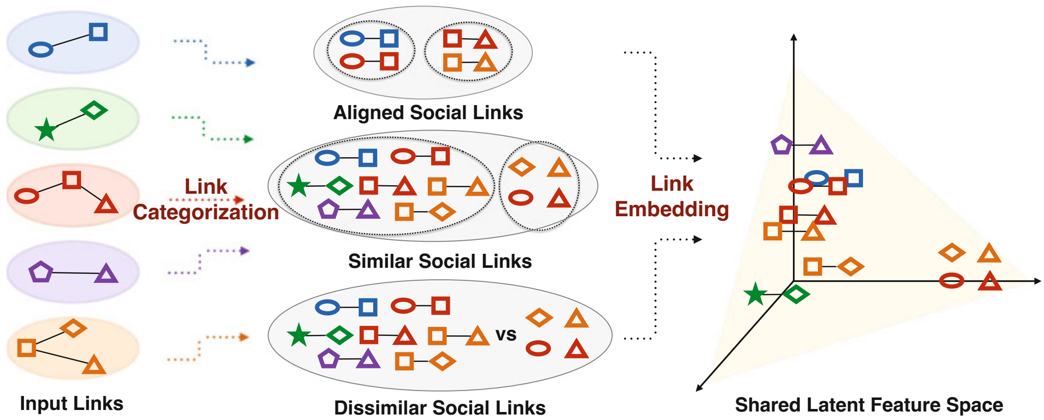


Fig. 7.13 An example of method SLAMPRED on social link embedding

SLAMPRED achieves the objective by utilizing the existing anchor links and social links across the networks. As shown in Fig. 7.13, the links in different social networks are first categorized into different sets: (1) social links aligned by anchor links (i.e., the *aligned social links* to be introduced later), (2) *similar social links* (i.e., connected user pairs or unconnected user pairs), and (3) *dissimilar social links* (i.e., the connected user pairs vs. the unconnected ones). Based on the categorization information about the links, in the link embedding process, we aim at placing aligned social links and similar social links closely in the common latent feature space, while placing the dissimilar ones far away from each other in the feature space. More information about these concepts and the embedding process will be introduced in the following parts in great detail.

7.6.3.1 Anchor Link Based Feature Space Projection

Before introducing the anchor link based feature space projection method, we first introduce the concept of *aligned social link* as follows:

Definition 7.6 (Aligned Social Link) Given two social links (u_i^t, u_j^t) and $(u_m^{(k)}, u_n^{(k)})$ in networks G^t and $G^{(k)}$, respectively, if $u_i^t, u_m^{(k)}$ and $u_j^t, u_n^{(k)}$ are both aligned by the anchor links (i.e., $(u_i^t, u_m^{(k)}) \in \mathcal{A}^{(t,k)}$ and $(u_j^t, u_n^{(k)}) \in \mathcal{A}^{(t,k)}$), then (u_i^t, u_j^t) and $(u_m^{(k)}, u_n^{(k)})$ are called the *aligned social links*.

Let sets $\mathcal{L}^t$ and $\mathcal{L}^{(k)}$ denote all the potential social links in networks G^t and $G^{(k)}$, respectively, where $\mathcal{L}^t = \mathcal{U}^t \times \mathcal{U}^t \setminus \{(u, u)\}_{u \in \mathcal{U}^t}$ and $\mathcal{L}^{(k)} = \mathcal{U}^{(k)} \times \mathcal{U}^{(k)} \setminus \{(u, u)\}_{u \in \mathcal{U}^{(k)}}$. Based on the anchor links between networks G^t and $G^{(k)}$ (i.e., $\mathcal{A}^{(t,k)}$), we can denote all the aligned social links with the *aligned social link indicator matrix* $\mathbf{W}_A^{(t,k)} \in \{0, 1\}^{|\mathcal{L}^t| \times |\mathcal{L}^{(k)}|}$, where entry $W_A^{(t,k)}(i, j) = 1$ iff the corresponding social links $l_i^t \in \mathcal{L}^t$ and $l_j^{(k)} \in \mathcal{L}^{(k)}$ are *aligned social links*.

Generally, the *aligned social links* are actually connecting the accounts of the same users, and the feature vectors extracted for them from different networks should be mapped to close areas in a low-dimensional latent feature space. Based on such an intuition, we can define the inconsistency introduced in projecting the features for aligned social links between networks G^t and other external source networks as term $Cost_A$:

$$Cost_A = \mu \sum_{m=t}^{K} \sum_{n=t}^{K} \sum_{i=1}^{|\mathcal{L}^{(m)}|} \sum_{j=1}^{|\mathcal{L}^{(n)}|} \left\| f^{(m)}(\mathbf{x}^m_{l_i^m}) - f^{(n)}(\mathbf{x}^n_{l_j^n}) \right\|^2 W_A^{(m,n)}(i,j), \tag{7.41}$$

where notation $\sum_{m=t}^{K}$ denotes the enumeration of all the networks in the set $\{G^t, G^{(1)}, \ldots, G^{(K)}\}$, and μ is the scalar.

Minimizing the cost term will encourage the features extracted for social links corresponding to the aligned social links being mapped to similar locations in the latent feature space. Furthermore, for all the pairwise networks, we can group all the *aligned social link indicator matrices* together as the big *joint aligned social link indicator matrix* $\mathbf{W}_A \in \{0,1\}^{|\mathcal{L}| \times |\mathcal{L}|}$, where $\mathcal{L} = \mathcal{L}^t \cup \mathcal{L}^{(1)} \cup \cdots \cup \mathcal{L}^{(K)}$. Formally, matrix $\mathbf{W}_A$ can be represented as

$$\mathbf{W}_A = \begin{bmatrix} \mathbf{W}_A^{(t,t)} & \mathbf{W}_A^{(t,1)} & \ldots & \mathbf{W}_A^{(t,K)} \\ \mathbf{W}_A^{(1,t)} & \mathbf{W}_A^{(1,1)} & \ldots & \mathbf{W}_A^{(1,K)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_A^{(K,t)} & \mathbf{W}_A^{(K,1)} & \ldots & \mathbf{W}_A^{(K,K)} \end{bmatrix}. \tag{7.42}$$

In addition, we can represent its Laplacian matrix as $\mathbf{L}_A = \mathbf{D}_A - \mathbf{W}_A$, where matrix $\mathbf{D}_A$ denotes the diagonal row sum matrix of $\mathbf{W}_A$ with entry $D_A(i,i) = \sum_j W_A(i,j)$ on the diagonal. Matrix $\mathbf{L}_A$ will be used in the projection function inference to be introduced in the following parts.

7.6.3.2 Existing Social Link Based Feature Space Projection

Besides the anchor link information, we also propose to utilize the existing social connections among the users to help infer the *feature mapping functions*. Before introducing the detailed method, we will define the concept of *link existence label* $y(\cdot)$ first as follows:

Definition 7.7 (Link Existence Label) Given a link $l_i^{(k)} \in \mathcal{L}^{(k)}$ in network $G^{(k)}, k \in \{t, 1, 2, \ldots, K\}$, if link $l_i^{(k)}$ exists in the network then its corresponding *link existence label* $y(l_i^{(k)}) = 1$, otherwise $y(l_i^{(k)}) = 0$.

Since our ultimate goal is to infer the potential feature vector mappings to the latent feature space to transfer information for the link prediction tasks, the social link existence information will plan a very important role in identifying the potential feature space mappings. Based on the known social connections in a pair of aligned networks G^t and $G^{(k)}$ ($k \in \{1, 2, \ldots, K\}$), we can construct the *similar link existence label indicator matrix* $\mathbf{W}_S^{(t,k)} \in \{0,1\}^{|\mathcal{L}^t| \times |\mathcal{L}^{(k)}|}$ and *dissimilar link existence label indicator matrix* $\mathbf{W}_D^{(t,k)} \in \{0,1\}^{|\mathcal{L}^t| \times |\mathcal{L}^{(k)}|}$ between networks G^t and $G^{(k)}$. For any link instances $l_i^t \in \mathcal{L}^t$ and $l_j^{(k)} \in \mathcal{L}^{(k)}$, if l_i^t and $l_j^{(k)}$ share the same *link existence label*, we will assign the corresponding entry in $\mathbf{W}_S^{(t,k)}$ with value 1 (and the corresponding entry in $\mathbf{W}_D^{(t,k)}$ with value 0); otherwise, we will assign the corresponding entry in $\mathbf{W}_S^{(t,k)}$ with value 0 (and the corresponding entry in $\mathbf{W}_D^{(t,k)}$ with value 1). Therefore, matrices $\mathbf{W}_S^{(t,k)}$ and $\mathbf{W}_D^{(t,k)}$ store all the link existence information in the networks G^t and $G^{(k)}$.

As pointed out in [44], the instances which share common labels tend to be projected together in the latent feature space, while those having different labels will be projected to be apart from each other instead. Based on such an intuition, terms $Cost_S$ and $Cost_D$ can be defined to denote the mapping costs introduced by the *link existence label* information (for the links having *similar* and *different* labels), respectively:

$$Cost_S = \sum_{m=t}^{K} \sum_{n=t}^{K} \sum_{i=1}^{|\mathcal{L}^{(m)}|} \sum_{j=1}^{|\mathcal{L}^{(n)}|} \left\| f^{(m)}(\mathbf{x}^m_{l^m_i}) - f^{(n)}(\mathbf{x}^n_{l^n_j}) \right\|^2 W_S^{(m,n)}(i, j), \tag{7.43}$$

$$Cost_D = \sum_{m=t}^{K} \sum_{n=t}^{K} \sum_{i=1}^{|\mathcal{L}^{(m)}|} \sum_{j=1}^{|\mathcal{L}^{(n)}|} \left\| f^{(m)}(\mathbf{x}^m_{l^m_i}) - f^{(n)}(\mathbf{x}^n_{l^n_j}) \right\|^2 W_D^{(m,n)}(i, j). \tag{7.44}$$

If link instances l_i^t and $l_j^{(k)}$ in networks G^t and $G^{(k)}$ share the same *link existence label* (i.e., $W_S^{(t,k)}(i, j) = 1$), but their embeddings are far away from each other, then $Cost_S$ will be larger. Meanwhile, if link instances l_i^t and $l_j^{(k)}$ have different *link existence labels* (i.e., $W_D^{(t,k)}(i, j) = 1$), and their embeddings are close to each other, the introduced $Cost_D$ will be small. Therefore, minimizing $Cost_S$ and maximizing $Cost_D$ simultaneously will encourage the link instances of the same label to be projected to similar areas, while those of different labels to be projected separately instead.

What's more, in a similar way, we can also group all the network pairwise *similar link existence label indicator matrices* and *dissimilar link existence label indicator matrices* together in the same order as matrix $\mathbf{W}_A$, which can be represented as $\mathbf{W}_S$ and $\mathbf{W}_D$. Their corresponding Laplacian matrices can be denoted as $\mathbf{L}_S$ and $\mathbf{L}_D$, respectively.

7.6.3.3 Joint Mapping Function Inference

We may want to ensure the mapping functions can achieve the above three objectives at the same time, which can be achieved by minimizing the overall cost function

$$\min Cost(f^t, f^{(1)}, f^{(2)}, \ldots, f^{(K)}) = \frac{Cost_A + Cost_S}{Cost_D}. \tag{7.45}$$

The projection mappings can be of different forms, and we will take the linear mapping as an example here. In other words, the mappings $f^t, f^{(1)}, f^{(2)}, \ldots, f^{(K)}$ can be represented as $K + 1$ matrices $\mathbf{F}^t \in \mathbb{R}^{d^t \times c}$, $\mathbf{F}^{(1)} \in \mathbb{R}^{d^{(1)} \times c}$, ..., $\mathbf{F}^{(K)} \in \mathbb{R}^{d^{(K)} \times c}$, respectively, where $d^t, d^{(1)}, \ldots, d^{(K)}$ denote the length of features from networks $G^t, G^{(1)}, \ldots, G^{(K)}$ and c is the dimension of the projected feature space.

Formally, given all the feature vectors extracted for potential user pairs in the networks $G^t, G^{(1)}, \ldots, G^{(K)}$, we can group them together and represent it as matrix

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}^t & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}^{(1)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{Z}^{(K)} \end{bmatrix}, \tag{7.46}$$

where submatrix $\mathbf{Z}^{(k)} = (\mathbf{z}_1^{(k)}, \mathbf{z}_2^{(k)}, \ldots, \mathbf{z}^{(k)}_{|\mathcal{L}^{(k)}| \times |\mathcal{L}^{(k)}|})$ and vector $\mathbf{z}_i^{(k)} \in \mathbb{R}^{d^{(k)} \times 1}$ represents the feature vector extracted for the i_{th} social link in network $G^{(k)}$. Furthermore, we can group all the projection function together and represent it as a $(d^t + d^{(1)} + \cdots + d^{(K)}) \times c$ dimensional matrix

$$\mathbf{F} = \left((\mathbf{F}^t)^\top, (\mathbf{F}^{(1)})^\top, \ldots, (\mathbf{F}^{(K)})^\top \right)^\top, \tag{7.47}$$

which can be effectively inferred with the following theorem.

Theorem 7.1 *The projection functions that minimize the overall cost function are given by the eigenvectors corresponding to the smallest non-zero eigenvalues of the generalized eigenvalue decomposition*

$$\mathbf{Z}(\mu \mathbf{L}_A + \mathbf{L}_S)\mathbf{Z}^\top \mathbf{x} = \lambda \mathbf{Z}\mathbf{L}_D\mathbf{Z}^\top \mathbf{x}. \tag{7.48}$$

Proof Depending on the specific value of c, the theorem can be proven by considering two cases:

Case 1 if $c > 1$, with the above defined matrices, we can rewrite the introduced cost terms $Cost_A$, $Cost_S$, and $Cost_D$ in the linear algebra representation:

$$Cost_A = \mathrm{Tr}(\mathbf{F}^\top \mathbf{Z}\mu \mathbf{L}_A \mathbf{Z}^\top \mathbf{F}), \tag{7.49}$$

$$Cost_S = \mathrm{Tr}(\mathbf{F}^\top \mathbf{Z}\mathbf{L}_S \mathbf{Z}^\top \mathbf{F}), \tag{7.50}$$

$$Cost_D = \mathrm{Tr}(\mathbf{F}^\top \mathbf{Z}\mathbf{L}_D \mathbf{Z}^\top \mathbf{F}). \tag{7.51}$$

Furthermore, the objective function can be represented as

$$\arg\min_{\mathbf{F}} \frac{\mathrm{Tr}(\mathbf{F}^\top \mathbf{Z}(\mu \mathbf{L}_A + \mathbf{L}_S)\mathbf{Z}^\top \mathbf{F})}{\mathrm{Tr}(\mathbf{F}^\top \mathbf{Z}\mathbf{L}_D \mathbf{Z}^\top \mathbf{F})}. \tag{7.52}$$

According to [44, 46], the matrix **F** which can minimize the objective function are actually the c eigenvectors corresponding to the c smallest non-zero eigenvalues of the following generalized eigenvalue decomposition function:

$$\mathbf{Z}(\mu \mathbf{L}_A + \mathbf{L}_S)\mathbf{Z}^\top \mathbf{x} = \lambda \mathbf{Z}\mathbf{L}_D\mathbf{Z}^\top \mathbf{x}. \tag{7.53}$$

Case 2 if $c = 1$, then matrix **F** to be inferred is actually a vector and the cost terms can be simply represented as

$$Cost_A = \mathbf{F}^\top \mathbf{Z}\mu \mathbf{L}_A \mathbf{Z}^\top \mathbf{F}, \tag{7.54}$$

$$Cost_S = \mathbf{F}^\top \mathbf{Z}\mathbf{L}_S \mathbf{Z}^\top \mathbf{F}, \tag{7.55}$$

$$Cost_D = \mathbf{F}^\top \mathbf{Z}\mathbf{L}_D \mathbf{Z}^\top \mathbf{F}. \tag{7.56}$$

The optimization objective function can be rewritten with the new cost representations as

$$\arg\min_{\mathbf{F}} \frac{\mathbf{F}^\top \mathbf{Z}(\mu \mathbf{L}_A + \mathbf{L}_S)\mathbf{Z}^\top \mathbf{F}}{\mathbf{F}^\top \mathbf{Z}\mathbf{L}_D \mathbf{Z}^\top \mathbf{F}}, \tag{7.57}$$

which is actually the *Rayleigh quotient* of $(\mu \mathbf{L}_A + \mathbf{L}_S)$ relative to $\mathbf{L}_D$. According to the existing books on linear algebra and related works [34, 38], the optimal solution to the objective function can be represented as the eigenvectors corresponding to the c small non-zero eigenvalues of the generalized eigenvalue problem:

$$\mathbf{Z}(\mu \mathbf{L}_A + \mathbf{L}_S)\mathbf{Z}^\top \mathbf{x} = \lambda \mathbf{Z}\mathbf{L}_D\mathbf{Z}^\top \mathbf{x}. \tag{7.58}$$

Therefore, we can formally represent the feature tensors of network G^k (including both the target and aligned source networks) after the domain adaption as $\hat{\mathbf{X}}^k \in \mathbb{R}^{|\mathcal{U}^k| \times |\mathcal{U}^k| \times c}$ ($\forall k \in \{t, 1, 2, \ldots, K\}$), where feature vector

$$\hat{\mathbf{X}}^k(i, j, :) = (\mathbf{F}^k)^\top \mathbf{X}^k(i, j, :). \tag{7.59}$$

7.6.3.4 Inter-Network Link Prediction Objective Function

With the information from the external source networks, we can obtain more knowledge about the users and their social patterns. Based on the adapted feature tensors $\hat{\mathbf{X}}^{(1)}, \ldots, \hat{\mathbf{X}}^{(K)}$, we can represent the intimacy scores of the potential social links as

$$int(\mathbf{S}, \hat{\mathbf{X}}^{(1)}, \ldots, \hat{\mathbf{X}}^{(K)}) = \sum_{k=1}^{K} \alpha^{(k)} \cdot int(\mathbf{S}, \hat{\mathbf{X}}^{(k)}) \tag{7.60}$$

where term $int(\mathbf{S}, \hat{\mathbf{X}}^{(k)}) = \left\| \mathbf{S} \circ \hat{\mathbf{X}}^{(k)} \right\|_1$, and users in $\hat{\mathbf{X}}^{(k)}$ are organized in the same order as $\mathbf{X}^t$. Parameters $\alpha^{(i)}$ denote the importance of the information transferred from the source network $G^{(i)}$. Furthermore, by adding the intimacy terms about the source networks into the objective function, we can rewrite it as follows:

$$\arg\min_{\mathbf{S} \in \mathcal{S}} \; l(\mathbf{S}, \mathbf{A}^t) - \alpha^t \cdot int(\mathbf{S}, \hat{\mathbf{X}}^t) - \sum_{k=1}^{K} \alpha^{(i)} \cdot int(\mathbf{S}, \hat{\mathbf{X}}^{(k)})) + \gamma \left\| \mathbf{S} \right\|_1 + \tau \left\| \mathbf{S} \right\|_* \tag{7.61}$$

7.6.4 Proximal Operator Based CCCP Algorithm

By studying the objective function, we observe that the intimacy terms are convex while the empirical loss term $l(\mathbf{S}, \mathbf{A}^t)$ is non-convex, which can be approximated with other classical loss functions (e.g., the hinge loss and the Frobenius norm), and the convex squared Frobenius norm loss function is used (i.e., $l(\mathbf{S}, \mathbf{A}^t) = \left\| \mathbf{S} - \mathbf{A}^t \right\|_F^2$). Therefore, the above objective function can be represented as a convex loss term minus another convex term together with two convex non-differentiable regularizers, which actually renders the objective function non-trivial. According to the existing works [37,51], this kind of objective function can be addressed with the concave-convex procedure (CCCP). CCCP is a majorization-minimization algorithm that solves the difference of convex functions problems as a sequence of convex problems. Meanwhile, the regularization terms can be effectively handled with the proximal operators [29] in each iteration of the CCCP process.

7.6.4.1 CCCP Algorithm

Formally, we can decompose the objective function into two convex functions:

$$u(\mathbf{S}) = l(\mathbf{S}, \mathbf{A}^t) + \gamma \cdot \left\| \mathbf{S} \right\|_1 + \tau \cdot \left\| \mathbf{S} \right\|_*, \tag{7.62}$$

$$v(\mathbf{S}) = \alpha^t \cdot int(\mathbf{S}, \hat{\mathbf{X}}^t) + \sum_{k=1}^{K} \alpha^{(k)} \cdot int(\mathbf{S}, \hat{\mathbf{X}}^{(k)}), \tag{7.63}$$

With $u(\mathbf{S})$ and $v(\mathbf{S})$, we can rewrite the objective function to be

$$\arg\min_{\mathbf{S}\in\mathcal{S}} u(\mathbf{S}) - v(\mathbf{S}). \tag{7.64}$$

The CCCP algorithm can address the objective function with an iterative procedure that solves the following sequence of convex problems:

$$\mathbf{S}^{(h+1)} = \arg\min_{\mathbf{S}\in\mathcal{S}} u(\mathbf{S}) - \mathbf{S}^\top \nabla v(\mathbf{S}^{(h)}) \tag{7.65}$$

It is easy to show that function $v(\mathbf{S})$ differentiable, and the derivative of function $v(\mathbf{S})$ is actually a constant term

$$\nabla v(\mathbf{S}) = \sum_{k=t}^{K} \alpha^{(i)} \sum_{i=1}^{c} \hat{\mathbf{X}}^{(k)}(i, :, :). \tag{7.66}$$

By relying on the Zangwill's global convergence theory [52] of iterative algorithms, it is theoretically proven in [37] that as such a procedure continues, the generated sequence of the variables $\{\mathbf{S}^{(h)}\}_{h=0}^{\infty}$ will converge to some stationary points $\mathbf{S}_*$ in the inference space $\mathcal{S}$.

7.6.4.2 Proximal Operators

Meanwhile, in each iteration of the CCCP updating process, objective function is not easy to address due to the non-differentiable regularizers. Some works have been done to deal with the objective function involving non-smooth functions. The Forward-Backward splitting method proposed in [8] can handle such a kind of optimization function with one single non-smooth regularizer based on the introduced proximal operators. More specifically, as introduced in [8], we can represent the proximal operators for the trace norm and L_1 norm as follows:

$$\text{prox}_{\tau\|\cdot\|_*}(\mathbf{S}) = \mathbf{U}\text{diag}((\sigma_i - \tau)_+)_i \mathbf{V}^\top, \tag{7.67}$$

$$\text{prox}_{\gamma\|\cdot\|_1}(\mathbf{S}) = \text{sgn}(\mathbf{S}) \circ (|\mathbf{S}| - \gamma)_+, \tag{7.68}$$

where $\mathbf{S} = \mathbf{U}\text{diag}(\sigma_i)_i \mathbf{V}^\top$ denotes the singular decomposition of matrix $\mathbf{S}$, and $\text{diag}(\sigma_i)_i$ represents the diagonal matrix with values σ_i on the diagonal.

Recently, some works have proposed the generalized Forward-Backward algorithm to tackle the case with $q (q \geq 2)$ non-differentiable convex regularizers [30]. These methods alternate the gradient step and the proximal steps to update the variables. For instance, given the above objective function in iteration h of the CCCP, we can represent the alternative updating equations in step k to address the objective function as follows:

$$\begin{cases} \mathbf{S}^{(k)} &= \mathbf{S}^{(k-1)} - \theta \cdot \nabla_{\mathbf{S}} \left(l(\mathbf{S}, \mathbf{A}) - \mathbf{S}^\top \nabla v(\mathbf{S}^{(h)}) \right), \\ \mathbf{S}^{(k)} &= \text{prox}_{\theta\tau\|\cdot\|_*}(\mathbf{S}^{(k)}), \\ \mathbf{S}^{(k)} &= \text{prox}_{\theta\gamma\|\cdot\|_1}(\mathbf{S}^{(k)}), \end{cases} \tag{7.69}$$

Algorithm 2 Proximal operator based CCCP algorithm

Require: social adjacency matrix $\mathbf{A}$
projected feature tensors $\hat{\mathbf{X}}^t, \hat{\mathbf{X}}^1, \ldots, \hat{\mathbf{X}}^K$
Ensure: link predictor matrix $\mathbf{S}$
Initialize matrix $\mathbf{S}_{cccp} = \mathbf{A}$
Initialize CCCP convergence CCCP-tag = False
while CCCP-tag == False **do**
  Initialize Proximal convergence Proximal-tag = False
  Solve optimization function $\min_{\mathbf{S} \in \mathcal{S}} u(\mathbf{S}) - \mathbf{S}^\top \nabla v(\mathbf{S}_{cccp})$
  Initialize $\mathbf{S}_{po} = \mathbf{S}_{cccp}$
  while Proximal-tag == False **do**
    $\mathbf{S}_{po} = \mathbf{S}_{po} - \theta \nabla_{\mathbf{S}} \left(l(\mathbf{S}_{po}, \mathbf{A}) - \mathbf{S}_{po}^\top \nabla v(\mathbf{S}_{cccp}) \right)$
    $\mathbf{S}_{po} = \text{prox}_{\theta\tau\|\cdot\|_*}(\mathbf{S}_{po})$
    $\mathbf{S}_{po} = \text{prox}_{\theta\gamma\|\cdot\|_1}(\mathbf{S}_{po})$
    if $\mathbf{S}_{po}$ converges **then**
      Proximal-tag = True
      $\mathbf{S}_{cccp} = \mathbf{S}_{po}$
    end if
  end while
  if $\mathbf{S}_{cccp}$ converges **then**
    CCCP-tag = True
  end if
end while
Return $\mathbf{S}_{cccp}$

where the parameter θ denotes the learning rate and it is assigned with a very small value to ensure the converge of the above functions [32]. We will also give the convergence analysis about the model in the experiment section.

The pseudo-code of the Proximal Operators based CCCP algorithm is available in Algorithm 2.

7.7 Summary

In this chapter, we introduced the link prediction problem in social networks, where various social network services can all be cast as the link prediction problem for simplicity. To address the problem, we introduced the traditional link prediction models for one single homogeneous networks, including the unsupervised link prediction models, supervised link prediction models, and the matrix factorization based link prediction models.

We also introduced the collective link prediction model for heterogeneous social networks, where we took the location-based social networks as an example to describe the problem setting and the proposed model. In the studied problem, we aimed at inferring multiple types of links in the location-based social networks simultaneously, including both the social links and location links. We provided a brief introduction of an integrated link prediction framework, which integrates these sub-problems into one unified framework.

To address the cold start problem in predicting potential links of new users, we introduced the cold start link prediction model, which can be built by utilizing the information about the "old users" within and across the networks. To accommodate the information distribution difference problem about the new users and old users, we introduced a method to sample the old users' subnetwork. Features extracted from multiple aligned heterogeneous can promisingly resolve the cold start problem in the proposed model.

Instead of modeling the non-existing links as the negative instances, we introduced an approach to address the link prediction problem as a PU learning problem, where those non-existing anchor links are treated as unlabeled instead. To identify a subset of the unlabeled anchor links which are highly likely to be negative (i.e., the reliable negative instances), we introduced to apply the spy techniques in the introduced model, which can work well to infer the social links in multiple networks concurrently.

To overcome the domain difference problem, at the end of this chapter, we introduced an approach to address the link prediction as a PU learning problem, where the link representations from different networks are projected into a shared low-dimensional feature space. Considering that the social network structure formed by the users are usually very sparse and users tend to form some small groups inside the social networks, the adjacency matrix of the social networks can have both the sparsity and low-rank properties. The introduced model resolves the problem as an optimization problem, where CCCP and proximal operators are adopted to learn the potential social links among the users.

7.8 Bibliography Notes

Link prediction problems is a traditional research problem studied in various areas, which aims at inferring the connections among nodes in the graph. To this context so far, dozens of link prediction works have been published already [3, 6, 9, 25, 47]. Depending on the learning setting utilized, the existing link prediction models for information networks can be divided into several categories. Initially, researchers study the link prediction problem based on an unsupervised learning setting [22], which predicts links by calculating the similarity scores among nodes with the assumption that close nodes are more likely to be connected. Afterwards, to utilize the supervision information and incorporate multiple closeness measures altogether, researchers introduce the supervised classification based link prediction models [14], where the existing and non-existing links are labeled as the positive and negative instances, respectively. Recently, researchers point that labeling the non-existing as negative instances is not reasonable, since some of the links will be formed, which should be unlabeled actually [54, 59]. Based on such an intuition, link prediction framework based on PU (Positive and Unlabeled) learning setting is introduced in [54, 59].

Most existing works solve link prediction problem with a single source of information. Nowadays, the researchers have pushed the problem boundary further forward by proposing the link prediction across multiple domains. Tang et al. [41] focus on inferring the particular type of links over multiple heterogeneous networks and develop a framework for classifying the type of social ties. To deal with the differences in information distributions of multiple networks, Qi et al. [3] propose to use biased cross-network sampling to do link prediction across networks. Meanwhile, some works have also been done on predicting multiple kinds of links simultaneously. Konstas et al. [20] propose to recommend multiple kinds of links with collaborative filtering methods. Fouss et al. [12] propose to use a traditional model, random walk, to predict multiple kinds of links simultaneously in networks.

Since Zhang et al. [19, 57] propose the concept of "*aligned social networks*," "*anchor links*," "*anchor users*," the social network studies across *multiple aligned social networks* have become a hot research area in recent years. Dozens of papers have been published around various problems about the multiple aligned networks, including *network alignment* [19, 55, 56] and *link prediction* [54, 57–60]. The link prediction models introduced in [54, 57–59] propose to combine the information from different sites by simply merging the extracted feature vectors together without considerations about the domain differences at all, which are totally different from the model introduced in this chapter. The recent paper [60] aims at unifying the link prediction problems subject to different cardinality constraints, like *one-to-one*, *one-to-many* and *many-to-many*, and introduce a general scalable link prediction framework to solve the problem.

To gain a more comprehensive knowledge about existing link prediction works, please refer to the survey paper [13, 35, 53] for more information.

7.9 Exercises

1. (Easy) Please implement the various unsupervised link predictors introduced in Sect. 7.2.1.2 and compare their effectiveness in inferring the friendship links within an online social network.
2. (Easy) Please explain the advantages of the supervised link prediction model over the unsupervised link predictors based on the closeness measures, e.g., *common neighbor* and *Jaccard's coefficient*.
3. (Easy) Please explain why the SCAN model introduced in Sect. 7.4 can resolve the *cold start problem* in predicting links for new users.
4. (Easy) Please define several of *inter-network meta paths* across aligned networks, and explain their physical meanings.
5. (Medium) Please try to implement the supervised link prediction model, and evaluate its performance with a synthetic network dataset.
6. (Medium) Please explain why the spy technique can help identify a set of *reliable negative* instances from the unlabeled set.
7. (Medium) Please explain why the L_1-norm and trace-norm introduced in Sect. 7.6 can maintain the sparse and low-rank properties of the matrix to be estimated.
8. (Hard) Please implement the matrix factorization based link prediction model introduced in Sect. 7.2.3, and evaluate its performance on a synthetic network dataset.
9. (Hard) Please implement the spy technique based PU learning algorithm introduced in Sect. 7.5.4, and use in the link prediction task.
10. (Hard) Please try to implement the sparse and low-rank matrix estimation based link prediction model introduced in Algorithm 2 with a preferred programming language.

References

1. K. Aditya, A. Menon, C. Elkan, Link prediction via matrix factorization, in *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II (ECML PKDD'11)* (2011)
2. C. Aggarwal, A. Hinneburg, D. Keim, On the surprising behavior of distance metrics in high dimensional spaces, in *Proceedings of the 8th International Conference on Database Theory (ICDT '01)* (2001)
3. C. Aggarwal G. Qi, T. Huang, Link prediction across networks by biased cross-network sampling, in *2013 IEEE 29th International Conference on Data Engineering (ICDE)* (2013)
4. A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, F. Li, S. Savarese, Social LSTM: human trajectory prediction in crowded spaces, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
5. M. Bilgic, G. Namata, L. Getoor, Combining collective classification and link prediction, in *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)* (2007)
6. B. Cao, N. Liu, Q. Yang, Transfer learning for collective link prediction in multiple heterogeneous domains, in *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10)* (2010)
7. E. Cho, S. Myers, J. Leskovec, Friendship and mobility: user movement in location-based social networks, in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)* (2011)
8. P.L. Combettes, V. Wajs, Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. **4**(4), 1168–1200 (2005)

9. Y. Dong, J. Tang, S. Wu, J. Tian, N. Chawla, J. Rao, H. Cao, Link prediction and recommendation across heterogeneous social networks, in *Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM '12)* (2012)
10. D. Dunlavy, T. Kolda, E. Acar, Temporal link prediction using matrix and tensor factorizations, in *ACM Transactions on Knowledge Discovery from Data (TKDD)* (2011)
11. C. Elkan, K. Noto, Learning classifiers from only positive and unlabeled data, in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)* (2008)
12. F. Fouss, A. Pirotte, J. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans. Knowl. Data Eng. **19**, 355–369 (2007)
13. M. Hasan, M.J. Zaki, A survey of link prediction in social networks, in *Social Network Data Analytics*, ed. by C.C. Aggarwal (2011)
14. M. Hasan, V. Chaoji, S. Salem, M. Zaki, Link prediction using supervised learning, in *Proceedings of SDM 06 Workshop on Link Analysis, Counterterrorism and Security* (2006)
15. W. Hsu, A. King, M. Paradesi, T. Pydimarri, T. Weninger, Collaborative and structural recommendation of friends using weblog-based social network analysis, in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs* (2006)
16. A. Huang, Similarity measures for text document clustering, in *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand* (2008)
17. P. Jaccard, Étude comparative de la distribution florale dans une portion des alpes et des jura. Bulletin del la Société Vaudoise des Sciences Naturelles **37**, 547–579 (1901)
18. K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. **20**(4) (2002)
19. X. Kong, J. Zhang, P. Yu, Inferring anchor links across multiple heterogeneous social networks, in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13)* (2013)
20. I. Konstas, V. Stathopoulos, J.M. Jose, On social networks and collaborative recommendation, in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)* (2009)
21. H. Kwak, C. Lee, H. Park, S. Moon, What is twitter, a social network or a news media? in *Proceedings of the 19th International Conference on World Wide Web (WWW '10)* (2010)
22. D. Liben-Nowell, J. Kleinberg, The link prediction problem for social networks, in *Proceedings of the Twelfth International Conference on Information and Knowledge Management (CIKM '03)* (2003)
23. D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. **58**(7) (2007)
24. B. Liu, Y. Dai, X. Li, W. Lee, P. Yu, Building text classifiers using positive and unlabeled examples, in *Proceedings of the Third IEEE International Conference on Data Mining (ICDM '03)* (2003)
25. Z. Lu, B. Savas, W. Tang, I. Dhillon, Supervised link prediction using multiple sources, in *2010 IEEE International Conference on Data Mining* (2010)
26. C. Lynch, Identifiers and their role in networked information applications. Bull. Am. Soc. Inf. Sci. Technol. **24**(2), 17–20 (2005)
27. M. McPherson, L. Smith-Lovin, J. Cook, Birds of a feather: homophily in social networks. Annu. Rev. Sociol. **27**, 415–444 (2001)
28. S. Pan, Q. Yang, A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**, 1345–1359 (2010)
29. N. Parikh, S. Boyd, Proximal algorithms. Found. Trends Optim. **1**(3), (2014)
30. H. Raguet, J. Fadili, G. Peyré, A generalized forward-backward splitting. SIAM J. Imag. Sci. (2013)
31. J. Ramos, Using TF-IDF to determine word relevance in document queries, in *Proceedings of the First Instructional Conference on Machine Learning* (1999), pp. 133–142
32. E. Richard, P. Savalle, N. Vayatis, Estimation of simultaneously sparse and low rank matrices, in *Proceedings of the 29th International Coference on Machine Learning (ICML'12)* (2012)
33. S. Scellato, A. Noulas, C. Mascolo, Exploiting place features in link prediction on location-based social networks, in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)* (2011)
34. J. Shi, J. Malik, Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
35. C. Shi, Y. Li, J. Zhang, Y. Sun, P. Yu, A survey of heterogeneous information network analysis. *CoRR*, abs/1511.04854 (2015)
36. E. Simpson, Measurement of diversity, in *Nature* (1949)
37. B. Sriperumbudur, G. Lanckriet, On the convergence of concave-convex procedure, in *Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS'09)* (2009)
38. Y. Sun, A. Damle, Matrix analysis notes. http://web.stanford.edu/~damle/refresher/notes/matrixanalysis.pdf

39. Y. Sun, R. Barber, M. Gupta, C.C. Aggarwal, J. Han, Co-author relationship prediction in heterogeneous bibliographic networks, in *2011 International Conference on Advances in Social Networks Analysis and Mining* (2011)
40. Y. Sun, J. Han, X. Yan, P. Yu, T. Wu, Pathsim: meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of 2011 International Conference on Very Large Data Bases (VLDB'11)* (2011)
41. J. Tang, T. Lou, J. Kleinberg, Inferring social ties across heterogeneous networks, in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM '12)* (2012)
42. J. Tang, H. Gao, X. Hu, H. Liu, Exploiting homophily effect for trust prediction, in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13)* (2013)
43. K. Torkkola, Feature extraction by non parametric mutual information maximization. J. Mach. Learn. Res. **3** (2003)
44. C. Wang, S. Mahadevan, Heterogeneous domain adaptation using manifold alignment, in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two (IJCAI'11)* (2011)
45. D. Wang, D. Pedreschi, C. Song, F. Giannotti, A. Barabasi, Human mobility, social ties, and link prediction, in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)* (2011)
46. S. Wilks, *Mathematical Statistics* (Wiley, Hoboken, 1963)
47. W. Xi, B. Zhang, Z. Chen, Y. Lu, S. Yan, W. Ma, E. Fox, Link fusion: a unified link analysis framework for multi-type interrelated data objects, in *13th International Conference on World Wide Web* (2004), pp. 319–327
48. M. Ye, P. Yin, W. Lee, Location recommendation for location-based social networks, in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)* (2010)
49. M. Ye, D. Shou, W. Lee, P. Yin, K. Janowicz, On the semantic annotation of places in location-based social networks, in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)* (2011)
50. J. Ye, H. Cheng, Z. Zhu, M. Chen, Predicting positive and negative links in signed social networks by transfer learning, in *Proceedings of the 22nd International Conference on World Wide Web (WWW '13)* (2013)
51. A. Yuille, A. Rangarajan, The concave-convex procedure. Neural Comput. **15**(4), 915–936 (2003)
52. W. Zangwill, *Nonlinear Programming* (Prentice-Hall, Englewood Cliffs, 1969)
53. J. Zhang, P. Yu, Link prediction across heterogeneous social networks: a survey (2014)
54. J. Zhang, P. Yu, Integrated anchor and social link predictions across partially aligned social networks, in *Twenty-Fourth International Joint Conference on Artificial Intelligence* (2015)
55. J. Zhang, P. Yu, Multiple anonymized social networks alignment, in *2015 IEEE International Conference on Data Mining* (2015)
56. J. Zhang, P. Yu, PCT: partial co-alignment of social networks, in *Proceedings of the 25th International Conference on World Wide Web (WWW '16)* (2016)
57. J. Zhang, X. Kong, P. Yu, Predicting social links for new users across aligned heterogeneous social networks, in *2013 IEEE 13th International Conference on Data Mining* (2013)
58. J. Zhang, X. Kong, P. Yu, Transferring heterogeneous links across location-based social networks, in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM '14)* (2014)
59. J. Zhang, P. Yu, Z. Zhou, Meta-path based multi-network collective link prediction, in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)* (2014)
60. J. Zhang, J. Chen, J. Zhu, Y. Chang, P. Yu, Link prediction with cardinality constraints, in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)* (2017)
61. J. Zhang, J. Chen, S. Zhi, Y. Chang, P. Yu, J. Han, Link prediction across aligned networks with sparse low rank matrix estimation, in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)* (2017)
62. E. Zheleva, L. Getoor, J. Golbeck, U. Kuter, Using friendship ties and family circles for link prediction, in *Advances in Social Network Mining and Analysis*, ed. by L. Giles, M. Smith, J. Yen, H. Zhang (2010)
63. K. Zhou, S. Yang, H. Zha, Functional matrix factorizations for cold-start recommendation, in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information (SIGIR '11)* (2011)