Breaking the Barrier to Transferring Link Information across Networks

Guo-Jun Qi, Charu C. Aggarwal, *Fellow, IEEE*, and Thomas S. Huang, *Life Fellow, IEEE*

Abstract—Link prediction is one of the most fundamental problems in graph modeling and mining. It has been studied in a wide range of scenarios, from uncovering missing links between different entities in databases, to recommending relations between people in social networks. In this problem, we wish to predict unseen links in a growing target network by exploiting existing structures in source networks. Most of the existing methods often assume that abundant links are available in the target network to build a model for link prediction. However, in many scenarios, the target network may be too sparse to enable robust inference process, which makes link prediction challenging with the paucity of link data. On the other hand, in many cases, other (more densely linked) auxiliary networks can be available that contains similar link structure relevant to that in the target network. The linkage information in the existing networks can be used *in conjunction with the node attribute information* in both networks in order to make more accurate link recommendations. Thus, this paper proposes the use of learning methods to perform link inference by transferring the link information from the source network to the target network. We also note that the source network may contain the link information irrelevant to the target network. This leads to cross-network bias between the networks, which makes the link model built upon the source network misaligned with the link structure of the target network. Therefore, we re-sample the source network to rectify such cross-network bias by maximizing the cross-network relevance measured by the node attributes, as well as preserving as rich link information as possible to avoid the loss of source link structure caused by the re-sampling algorithm. The link model based on the re-sampled source network can make more accurate link predictions on the target network with aligned link structures across the networks. We present experimental results illustrating the effectiveness of the approach.

Index Terms—Link prediction, link transfer, cross-network bias, node attribution, link richness

1 Introduction

The goal of link inference in a certain network is to predict links between nodes based on its current structure and the content in the nodes [1], [2], [3], [4], [5], [6], [7], [8]. Most of the existing techniques predict the links in a growing network based on its local structure. They often assume that two nodes are more likely to be linked, if they are structurally reachable through some existing nodes. For example, in many social networks, link predictions are made between two nodes, when the two nodes have common contacts in the networks.

The structural information in a network has been proven extremely powerful and reliable for link prediction [7]. Useful structural information includes the number of common neighbors (CN), and the length of the shortest path between the nodes. However, such an approach becomes vulnerable if insufficient structural information is available in an "infant" network with sparse links, because a given pair of nodes may not have a lot of common neighbors or be connected by a short path, even when they are closely related to one another. Therefore, the traditional methods for link inference can fail because of the paucity of available structural information. The link inference problem is particularly important for the sparse (or new) networks to grow, whose basic structure of the networks is not known to a large degree. *Therefore, the existing methods for link inference are particularly challenged in scenarios where link inference plays the critical role for recommending relationships and growing social networks.*

In this paper, we will develop the cross-network link prediction (CNLP) model, which attempts to leverage the existing link information in a mature source network (e.g., *Facebook*) to predict the links in a relatively new network (e.g., *Google+*). The link inference problem can also be treated as a classification problem in which the derived features between node pairs, such as structural connections or attribute correlations, can be used to build prediction models on the training data of which the existences of links are known. They can then be applied to the target network to predict the potential links between node pairs. The connection between the link inference problem and the classification problem motivates a natural approach in which both the structural and attribute information are extracted in order to enhance the link inference model. This is related to the problem of *transfer learning* [9], which is widely used to mitigate the problem with paucity of data in one domain by incorporating rich auxiliary information from the other domains.

Social networks typically contain a rich amount of content attributes at the nodes, which can be used as a bridge in order to connect the linkage behavior of the two networks.

- *G.-J. Qi is with the Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816. E-mail: guojun.qi@ucf.edu.*
- *T. S. Huang is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801. E-mail: huang@ifp.uiuc.edu.*
- *C. C. Aggarwal is with the IBM T.J. Watson Research Center, Yorktown Heights, NY 10598. E-mail: charu@us.ibm.com.*

Manuscript received 28 Oct. 2013; revised 17 Feb. 2014; accepted 20 Feb. 2014. Date of publication 25 Mar. 2014; date of current version 1 June 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TKDE.2014.2313871

For example, if nodes containing the keyword *Copperbeach High School: Class of 1989* are highly connected to one another in the training network, it provides a hint that these nodes may also be connected in the second network. In such a case, the nodes may not exactly correspond to the same actors, but the keywords point to an interest that is highly related to linkage behavior. Therefore, the transfer process needs to *learn* the content that is most highly correlated to the linkage behavior. In combination with the *sparse linkage information* in the target network, it may be possible to significantly enhance the link inference process.

In some cases, a particular combination of keywords may be useful in order to predict links between nodes. For example, the training data may suggest that certain combinations of keywords in a pair of nodes may be highly indicative of a link, though the keywords may not exactly be the same. For example, the keywords {*IBM machine learning*} may be highly linked with nodes containing {*IBM data mining*}, though it may not be as closely related to nodes containing the keywords {*IBM Systems and Hardware*}. Thus, the precise combination of content needs to be *learned* from the training network in the transfer process. Furthermore, the sparse structural information can be more effectively used when such content information is available. For example, the presence of only one common friend between two nodes in the sparse target network may not constitute sufficient information for link inference, but the presence of the keywords {*IBM machine learning*} and {*IBM data mining*} may significantly enhance this probability.

In many cases, a *combination* of the content information in the training network and the (sparse) structure of the target network can be used in order to make effective inferences about the links in the target network.

1.1 Barrier to Transferring Link Information across Networks: Cross-Network Bias

In this paper we will develop a cross-network link prediction model by using the linkage information in the source network in order to predict links in the target network. This is different from traditional link inference, in which only the previous links of a single network may be used for its future link prediction. A natural challenge inherent in such an approach is that the two networks are distinct, and may even be drawn from different domains, such as a traditional social network and a bibliographic network. This implies that the source and target networks may be generated from very different distributions. Even in cases, where the networks are drawn from similar domains, there are likely to be inherent differences in the content and structure of the two networks. This leads to a significant amount of *cross-network bias*, which can be very detrimental to the transfer process, in the form of significant errors and over-fitting. Thus, a blind transfer process may not be very helpful for effective learning in the link inference process. In this paper, we propose a network re-sampling technique for carefully calibrating the portions of the source network to be used in the transfer process. This provides a bias-correction methodology, which is combined with a transfer learning-based linked prediction model for ensuring robust and effective link prediction.

This paper is organized as follows. In Section 2, we will review the related work and their limitations. Then we will formalize the cross-network link prediction problem and present the main ideas and challenges in Section 3. A link model is built on the source network in Section 4. In Section 5, a re-sampling process will be proposed to align the link structures between the source and target networks, so that the link information can be shared and transferred between the networks based on the link model in Section 4. We present experimental results in Section 6 to demonstrate the effectiveness of our approach. The conclusions and summary are presented in Section 7.

2 Related Work

The problem of link prediction has been studied extensively in the data mining and machine learning community [10]. Much of the work on this problem is based on defining proximity-based measures on the nodes in the underlying network [7], [11], [12]. The work in [7] studied the usefulness of different topological features for link prediction. It was discovered in [2] that none of the features was particularly dominant in different kinds of situations. A second approach is to study the problem in the context of statistical relational models [3], [13], [14], [15], [16], [17].

The link prediction problem has also been studied more generally in the context of the classification problem [2], [5], [6]. Specifically, the existence of an edge between a pair of nodes can be considered a binary class label, which can be predicted with the use of either derived or existing attributes between the pair of nodes. For example, the similarity in content-attributes (existing textual information), and the similarity in structural neighbors correspond to derived attributes, which can be used for link inference. Intuitively the larger the similarity between the node pair, the more likely a link will exist. It is possible to use the current set of links in order to create a training data set, which is used for link inference [2], [5], [6] for node pairs in which the presence or absence of links is unknown. The connections of the link inference problem with that of classification point to a natural approach of using transfer learning methods [18], [19] for transferring knowledge from mature networks with dense linkage behavior to the target (sparse) network in which a paucity of linkage information is a problem for the learning process.

Recently, some researchers have noticed and work on this link transfer problem. For example, a method has been proposed for labeling *already existing* edges in a social network with the use of labeling information from another network [20]. This is different from the problem of link prediction discussed in this paper, where the actual *existence* of a link needs to be predicted.

Ye et al. [21] also proposes a transfer learning algorithm for link prediction. This method constructs latent topological features in a shared space on both source and target networks. While it attempts to integrate the linking information in both networks, it is risky of involving the irrelevant structures in source network in predicting the links in target network. On the contrary, we explicitly re-sample the most relevant structures in the source network, which

can mitigate the cross-network bias when transferring the linking information.

In addition, [22] develops a de-anonymization method for link prediction by finding the correspondence between networks. It de-anonymizes a target network by weighted graph matching with the help of an efficient simulated annealing method. Compared with the problem they addressed, we instead focus on a more challenging one, where the source end target networks come from completely different social media platforms. That means the graph matching technique may fail as their topological structures almost completely differ from one another. That is why we propose to sort to the content information that reveals important auxiliary information for link transfer, in addition to the topological features used in most of existing link prediction models.

3 Overview

In this section, we will define the link inference problem, as it relates to *cross-network transfer learning*. We denote the source network by $\mathcal{G}^0 = (\mathcal{V}^0, \mathcal{E}^0)$, and the target network by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The links need to be predicted in target network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that is assumed to be nascent and sparse. The node sets in the source and target networks are denoted by $\mathcal{V}^0 = \{v_1^0, v_2^0, \ldots, v_m^0\}$ and $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ respectively. Each edge is denoted by $(v_i^0, v_j^0) \in \mathcal{E}^0$ and $(v_s, v_t) \in \mathcal{E}$ respectively. For ease in notation, we will use different subscripts i, j and s, t to differentiate the nodes and edges in $\mathcal{G}^0$ and $\mathcal{G}$ respectively. The source network $\mathcal{G}^0$ is assumed to be a mature network of nodes and edges that has more linkage information than the target network $\mathcal{G}$. Thus, the source network $\mathcal{G}^0$ contains substantially more linkage and content knowledge, which can be leveraged for the link inference process. The correspondence between the nodes in $\mathcal{V}$ and $\mathcal{V}^0$ is unknown, and the only information that relates the nodes in $\mathcal{V}$ and $\mathcal{V}^0$ is the available attribute information at the nodes. In fact an exact correspondence may not even exist, especially since one of the networks is likely to be significantly larger than the other. In some cases, the networks may be pre-labeled with the actor name, though this does not necessarily provide exact correspondence, given the enormous ambiguities inherent in such labels. Such a label can at best be considered an attribute of the node, which can be used in order to help the transfer process of the link structure. In other cases, the networks may be anonymized, and only a limited amount of attribute information (such as keywords corresponding to the profile) may be available. However, the network does provide useful information about the nature of the attributes in the two networks that more likely to be linked together. We assume that each node in $\mathcal{V}$ (and $\mathcal{V}^0$) is associated with a set of keywords that are derived from the profile information in the two social networks. Specifically, we denote the attributes associated with the node $v_s \in \mathcal{G}$ and $v_i^0 \in \mathcal{G}^0$ by feature vectors $\mathbf{x}_s$ and $\mathbf{x}_i^0$ in the vector space $\mathbb{R}^d$ of dimension d. These keywords may include the actor name in cases where such information is available. The cross-network link prediction problem is defined as follows:

Problem 3.1 (Cross-network link prediction). *Given the training network $\mathcal{G}^0 = (\mathcal{V}^0, \mathcal{E}^0)$, along with its associated content attributes $\overline{\mathbf{x}^0}$, determine the links that have the highest probability to appear in the future in a currently existing target network $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with corresponding content attributes $\overline{x}$.*

3.1 Broad Intuition and Preliminaries

The task of cross-network link prediction is to leverage the link structure in the source network in order to predict the links in target network. In many previous works [5], [16], [15], [7], [6], the most direct approach is to train a link model from a given network in order to predict the *future* links *within the same network*. In some sense, the traditional link prediction problem can be considered a special case of cross-network link prediction, in which the target network is the same network as the source network at a future point in time. The cross-network link prediction considers much broader scenarios, where the source and target networks may have completely different sets of nodes. For example, the source and target networks could be distinct social networks, (such as *Google+* and *Facebook* networks), or they could correspond to co-authorship networks between authors from different research areas.

One of the crucial parts of link prediction process is to design a knowledge transfer relationship of the content at the different nodes with the linkage probability between the nodes. This knowledge is particularly useful for the facilitation of accurate inferences of the links among the nodes in the two networks. Of course, the relationship of the linkage probabilities to the node content may not be precisely identical between the two networks. For example, consider two co-authorship networks, which are focussed on the different topics of *information retrieval* and *web mining*. Although the researchers in these two networks have research interests and expertise in common, their underlying *distributions* in the two networks may be quite different. While a relatively larger number of the researchers in the information retrieval network may concentrate on *retrieval theory and models*, more researchers in the *web mining* network may be interested in *web search and mining*. Therefore, the same content may have different linkage relevance and distribution in different networks. This also means that the relationship of linkage structure to content may vary in the two networks to some extent. As the collaboration links are created based on the common research interests and expertise between authors, this implies that a direct transfer of the content-link relationships in the source network to the target network may not be very helpful. This is essentially a form of *cross-network bias* in the learning process. Therefore, we need to design methods that are robust to variations between the two networks. In order to achieve this goal, we will propose a cross-network transfer model, which uses a link-sampling parameter as an integral part of the model. Then, we will discuss how the cross-network bias can be eliminated with the use of careful sampling of the links during the transfer process.

In Fig. 1, we plot an overview of the proposed link transfer algorithm across networks. It consists of three main steps as follows:

1: Re-sampling the source network to rectify the cross-network bias. This can be accomplished by maximizing the cross-network relevance between the nodes

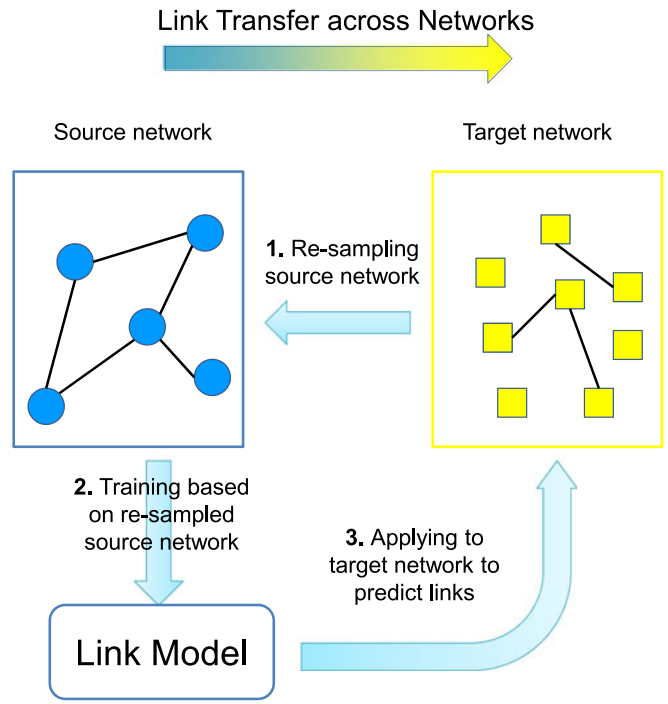


Fig. 1. An overview of the proposed cross-network link transfer algorithm–Step 1: re-sampling the source network to rectify the cross-network bias; Step 2: training the cross-network link prediction model based on the re-sampled source network, combined with the existing link information in the target network; Step 3: applying the learned link model to the target network to infer the unknown links.

in the source and target networks, subject to maximizing the loss of link information in social network.

2: Training the cross-network link prediction model based on the re-sampled source network, along with the existing link information in the target network.

3: Applying the learned link model to the target network to infer the unknown links.

In the following sections, we will explain the components of the algorithms step by step.

4 Cross-Network Link Model

In this section, we will show how to leverage the link information in the source network in order to predict the links in the target network. Associated with each link in the source network, we will define a sampling parameter P_{ij}, which essentially represents the importance of a link between the ith and jth nodes in the source network during the link transfer process. This is essentially a way of calibrating cross-network relevance during the link-transfer process. In this section, we will design a link-transfer model with the general use of this sampling parameter, without discussing how it is derived. In a later section, we will explicitly discuss how this parameter is actually determined by addressing the dual goals of cross-network bias correction and structural richness.

Our model for cross-network link transfer uses a latent space approach that relates the network attributes to the probability of link presence in the source and target networks. This is used in order to perform the knowledge transfer between the source and target networks. Furthermore, as the target network evolves over time, new links will be created between nodes and become available for learning in the target network. These links provide auxiliary knowledge about the link structure in the target network that are complementary to the link information in the source network. As in the case of traditional link prediction, such links can be used in order to improve the effectiveness of the transfer process.

Before discussing the model in detail, we will introduce some notations and definitions. The current target network is denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is its node set, and $(v_s, v_t) \in \mathcal{E}$ is an edge in $\mathcal{G}$. The attribute vector associated with each node v_s in the target network is denoted by $\mathbf{x}_s$. For example, in a co-authorship network, the attribute vector of an author node may correspond to the keywords of their published papers. In the context of a traditional social networks such as *Facebook* and *Twitter*, such attributes may correspond to the content of the posts and the profiles of the network actors. Similarly, we have a source network $\mathcal{G}^0 = \{\mathcal{V}^0, \mathcal{E}^0\}$, with analogous attribute information associated with the nodes. Since the information associated with the different nodes in the source and target networks are analogous, we consistently use the superscript "0" in all source network notations, in order to distinguish from the target network. Then, the link prediction in the target network can be solved by combining the link and content information in the source network $\mathcal{G}^0$ with the currently existing link and content information in the target network $\mathcal{G}$ in order to predict the future links in the latter network.

The link prediction problem can be formulated as a learning problem on the links and the associated node content [23]. Specifically, the content vectors $\mathbf{x}_i^0$ for each node v_i^0 in $\mathcal{G}^0$ and $\mathbf{x}_s$ for v_s in $\mathcal{G}$ are mapped to $\boldsymbol{\varphi}_i^0$ and $\boldsymbol{\varphi}_s$ in a latent topic space $\mathbb{R}^k$ respectively, by a linear transformation as $\boldsymbol{\varphi}_i^0 = \mathbf{W} \cdot \mathbf{x}_i^0$ and $\boldsymbol{\varphi}_s = \mathbf{W} \cdot \mathbf{x}_s$ with the $k \times d$ matrix $\mathbf{W}$. Here k is the dimension of the latent topic space, and the value of k can be chosen based on the Bayesian information criterion (BIC) [24]. It is assumed that the matrix $\mathbf{W}$ needs to be learned, and the goal of this learned topic space is to maximize the log likelihood of the link prediction probabilities of our content and structural model for link prediction. The social interaction between two nodes v_s and v_t in the target network can be measured by the inner product $\boldsymbol{\varphi}_s' \cdot \boldsymbol{\varphi}_t$ between the corresponding latent vectors. In other words, this social interaction measures the similarity between the content-based latent vectors associated with these two nodes. For example, the collaboration links between the authors in a co-authorship network can be inferred based on the similarity between the latent topic vectors of their research interests and expertise. Therefore, we will model the link prediction probabilities as a function of these similarity values [25], and then try to learn the precise function, which maximizes the log-likelihood probabilities. Thus, the matrix $\mathbf{W}$ plays a key role in the inference process, and it is critical to learn its optimal value in order to infer the links.

In addition to the link-attribute interaction, which is encoded in the associated content-based latent vectors, the topological features, such as common neighbors of the two nodes v_s and v_t and Adamic-Adar (AA) [11], provide useful topological hints to infer the future links in the network. In

this paper, we use the Adamic-Adar feature b_{st} defined on a pair of nodes v_s and v_t to capture the common neighbors in the target network. This feature is chosen for its effectiveness in modeling the local topological structure in the networks [11]. Specifically, AA feature measures the similarity between a pair of nodes in a network as the number of their common neighbors with different importance weighted by their rarity. The idea that is the common neighbors unique to fewer nodes are more effective in connecting people than those nodes with a larger group of neighbors [11]. Then, the probability that the two nodes v_s and v_t will be linked in the future is modeled as a combination function of the latent vector and structural (Adamic-Adar) components:

$$\Pr(y_{st} = +1|\mathcal{G}) = f(\boldsymbol{\varphi}_s' \cdot \boldsymbol{\varphi}_t + \alpha \cdot b_{st}), \tag{1}$$

where $y_{st} = +1$ indicates there will be a link between v_s and v_t in the future, and $y_{st} = -1$ indicates otherwise. We note that the sigmoid function $f(z) = 1/(1 + \exp(-z))$, which is used to represent the expression for $\Pr(y_{ij} = +1|\mathcal{G})$ will always lie in $[0, 1]$. The parameter $\alpha \geq 0$ is the combination coefficient, which determines the relative importance of the two terms. It is noteworthy, that the determination of matrix $\mathbf{W}$ directly yields the probabilities of the links $\Pr(y_{st} = +1|\mathcal{G})$, which can be directly used for link prediction. Therefore, it remains to discuss how the matrix $\mathbf{W}$ should be learned in an optimal way. We further note that while the above computation is performed on the target network, the matrix $\mathbf{W}$ is determined with the use of an optimally picked joint latent space in the source and existing target networks. This ensures that the link-prediction process encodes the knowledge available in the both networks for the transfer process.

The learning process for the matrix $\mathbf{W}$ tries to determine a topic space in which nodes with relevant content in them (based on source matrix connectivity), as well as nodes that are topologically well connected in the target network tend to be placed close together in the topic space. Specifically, the learning process contains two components in the objective function, which are used to perform the prediction:

- The current state of the (nascent) target network in terms of its content and structure, which may contain some information for link prediction.
- The cross-network knowledge that is transferred from the source to the target network.

In the following, we will design an objective function that contains components for both of the above, and learn the matrix $\mathbf{W}$, which maximizes the log-likelihood probabilities for link prediction. In order to learn the mapping for the latent space, we have the following logarithmic likelihood of the **existing** links in the target network $\mathcal{G}$:

$$\begin{aligned}\mathcal{L} &= \sum_{(v_s,v_t)\in\mathcal{E}} \log f(\varphi_s' \cdot \varphi_t + \alpha \cdot b_{st}) \\ &= -\sum_{(v_s,v_t)\in\mathcal{E}} \ell(\varphi_s' \cdot \varphi_t + \alpha \cdot b_{st}).\end{aligned} \tag{2}$$

We assume that $\ell(z) = \log(1 + \exp(-z))$ is the logistic loss function, and the corresponding maximum likelihood criterion is essentially equivalent to performing logistic regression (LR) on the variables corresponding to the existence of the network links. We note that the value $\mathcal{L}$ is the first component of the objective function that uses information only about the target network, without considering the cross-network information from the source network.

As mentioned earlier, the links in the current target network $\mathcal{G}$ are quite sparse in scenarios where the network is nascent, and it is not sufficient to either perform traditional link prediction, or to yield a robust enough latent topic space in which the social interactions between the nodes can be predicted. In contrast, the source network contains rich linkage information for learning the robust representation of latent topics. For this purpose, we combine the model with knowledge from a re-sampled source network based on a sampling importance of the link between nodes v_i^0 and v_j^0, denoted by P_{ij}. This forms the second component of our objective function, and can be written as follows:

$$\mathcal{L}^0 = -\sum_{v_i^0, v_j^0 \in \mathcal{V}^0} P_{ij} \cdot \ell\big(y_{ij}^0 \cdot \boldsymbol{\varphi}_i^{0\prime} \cdot \boldsymbol{\varphi}_j^0\big). \tag{3}$$

We assume that a link exists between v_i^0 and v_j^0 when $y_{ij}^0 = 1$, and otherwise when $y_{ij}^0 = -1$. The above equation equals the expected log likelihood of links over the sampled source network. This component in the objective function provides an effective transfer learning of the content-link relationships in the source network.

The parameter P_{ij} in Eq. (3) weighs the importance of sampling the link (v_i^0, v_j^0) in the source network. It is noteworthy that the importance weights P_{ij} play a crucial role in sampling the relevant link information in the source network for an effective transfer learning process. Due to the afore-mentioned cross-network bias, not all the links in the source network are generated from the same distribution underlying the target network. Therefore, if we equally weigh all the links in the source network, this can undermine the link transfer process between the networks. Therefore, in the next section, we present a method for re-sampling the source network to correct the cross-network bias. This provides the probability P_{ij}, which is used above.

By maximizing the combined log-likelihood of links in the source and target networks, we can learn the optimal latent transformation matrix $\mathbf{W}$:

$$\begin{aligned}\mathbf{W}^\star &= \arg\max_{\mathbf{W}} \; -\mathcal{L} - \eta\mathcal{L}^0 + \gamma\|\mathbf{W}\|_2^2 \\ &= -\sum_{(v_s,v_t)\in\mathcal{E}} \ell\big(\mathbf{x}_s'\mathbf{W}'\mathbf{W}\mathbf{x}_t + \alpha b_{st}\big) \\ &\quad - \eta \sum_{v_i^0, v_j^0 \in \mathcal{V}^0} P_{ij}\ell\big(y_{ij}^0 \mathbf{x}_i^{0\prime}\mathbf{W}'\mathbf{W}\mathbf{x}_j^0\big) + \gamma\|\mathbf{W}\|_2^2.\end{aligned} \tag{4}$$

The last term imposes a regularizer for better generalization performance, and η and γ are the balancing parameters trading off between the different terms in the objective function, which correspond to the cross-network information from the source, and the existing information in the target. The above objective function is differentiable with respect to the parameter $\mathbf{W}$, and can be efficiently solved by the off-the-shelf unconstrained optimization solver such as conjugate gradient method [26].

Once the optimal latent representation parameterized by $\mathbf{W}$ is learned from the above objective function, we can

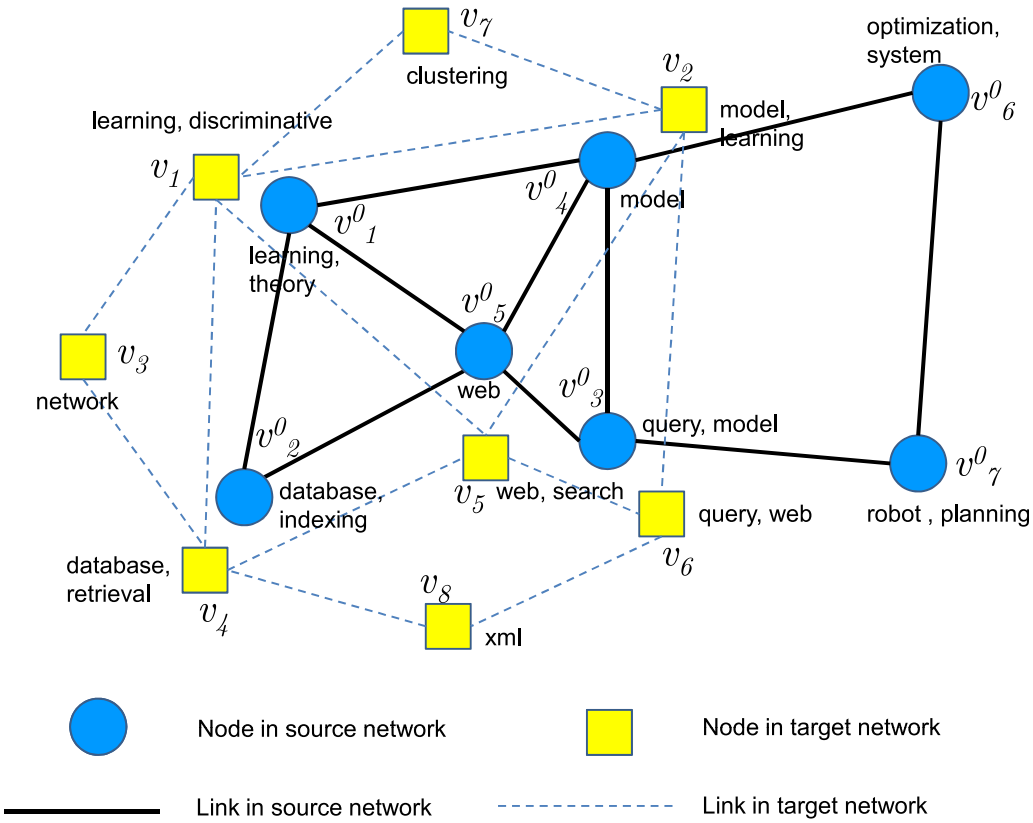


Fig. 2. An example of the bias in attribute information associated with source and target networks.

compute the value of the expression $\boldsymbol{\varphi}'_s \cdot \boldsymbol{\varphi}_t + \alpha \cdot b_{st}$. This expression can be used in conjunction with Eq. (1) in order to predict the probability of a link between a pair of nodes.

5 Cross-Network Bias Correction

In this section, we will discuss the determination of the sampling weights P_{ij} used in Eq. (3) of the last section, in order to correct for bias. The idea is to ensure that the links in the target network, which are consistent with the source network in terms of the node-content relationships are given much greater importance. At the same time, the re-sampling process also need to preserve the richness of link structure in the source network as much as possible, in order to maximize its utility in the learning process.

The existing distribution bias correction algorithms [9] on traditional *relational* data calibrate the *sample bias* between the training and test data sets, by minimizing the sample mean between the training and test data sets. However, such an approach is not designed for network structural data, in which the link information needs to be retained during the sampling process. Therefore, we present a new approach to correct the cross-network bias, which also preserves and transfers the link information in the source network to the target network.

Fig. 2 illustrates an example of cross-network bias between the source and target networks. For illustration, we have presented some important attributes associated with the nodes. The closer the two nodes are, the more relevant their attributes are to each other. It is evident that the nodes $\{v_1^0, v_2^0, v_3^0, v_4^0, v_5^0\}$ in the source network are more relevant to the nodes in the target network. Consequently, they provide more linking clues to the target network and they should have larger sampling weights than $\{v_6^0, v_7^0\}$ in the re-sampling process. On the other hand, the re-sampling process ought to preserve as much link information as possible as to minimize the lost link information in the source network. For example, in Fig. 2 the link structure between the relevant nodes $\{v_1^0, v_2^0, v_3^0, v_4^0, v_5^0\}$ should be kept intact to preserve the links incident with these relevant nodes. By correctly sampling these links and nodes, the obtained re-sampled source network provides a more robust template for the transfer process.

In this section, we will discuss the basics of the re-sampling process. The broad goal of this process is to achieve the following:

1. Maximize the consistency between the source and target networks in terms of the attributes associated with their nodes.
2. Preserve the richness of the structure of the sampled network, so that as much structural information as possible is available for the transfer learning process.

We provide quantifications of the afore-mentioned criteria, so that a concrete tradeoff may be obtained for creating the re-sampled network in the transfer process.

5.1 Re-Sampling the Source Network

In the re-sampled source network, each node is sampled in iid fashion according to a weighting distribution $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_n\}$ on the node set $\mathcal{V}^0$ of the source network, where

$\sum_{i=1}^{n} \beta_i = 1, \beta_i \geq 0$. Formally, we have the following definition for a re-sampled source network:

Definition. *A re-sampled source network $\overline{\mathcal{G}}^0 = \{\overline{\mathcal{V}}^0, \overline{\mathcal{E}}^0, \boldsymbol{\beta}\}$ is a stochastic network structure, whose nodes are sampled from the node set $\mathcal{V}^0$ of the source network $\mathcal{G}^0$ according to the sampling weights $\boldsymbol{\beta}$. Formally, a node U in the re-sampled network $\overline{\mathcal{G}}^0$ is a random variable that takes on values from $\overline{\mathcal{V}}^0$ with the probability that $\Pr(U = v_i^0) = \beta_i$ for $i = 1, 2, \ldots, n$.*

By the above definition, the probability P_{ij} of sampling a link $(v_i, v_j) \in \mathcal{E}^0$ in the re-sampling process can be computed as follows:

$$\begin{aligned} P_{ij} &= \Pr((U, W) = (v_i, v_j)) \\ &= \Pr(U = v_i, W = v_j) + \Pr(U = v_j, W = v_i) \\ &= \Pr(U = v_i) \cdot \Pr(W = v_j) + \Pr(U = v_j) \cdot \Pr(W = v_i) \\ &= 2\beta_i \cdot \beta_j. \end{aligned} \tag{5}$$

In the second equality, we assume that the two random variables U and W corresponding to the nodes are independently sampled from $\mathcal{V}^0$. This re-sampling probability P_{ij} of the links in the source network is the critical parameter that is require to complete the transfer learning model of the last section, according to Eq. (3).

Since the value of P_{ij} depends upon the sampling distribution $\boldsymbol{\beta}$, our goal is to determine the value of $\boldsymbol{\beta}$, which minimizes the cross-network bias, while retaining the richness in network structure. We will discuss the quantification of these goals in the following two sections, and the optimal determination of the distribution $\boldsymbol{\beta}$ on this basis.

5.1.1 Cross-Network Relevance

The relevance $R(\mathcal{G}^0, \mathcal{G})$ between the source and target networks measures the consistency of the distributions underlying these two networks. A naive method for computing the cross-network relevance without considering node distributions, would be to simply measure the average attribute similarity between the nodes of the networks. Such a naive definition of relevance would be as follows:

$$R(\mathcal{G}^0, \mathcal{G}) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{s=1}^{m} S(v_i^0, v_s). \tag{6}$$

Here, $S(v_i^0, v_s)$ is the similarity between the attributes of the nodes v_i^0 and v_s. These attributes may correspond to different kinds of content in different networks, such as the publication content in research networks, or the user-posted messages in social networks. In our paper, we use the cosine similarity as the similarity function $S(\cdot, \cdot)$. Ideally, if the nodes in the two networks are generated from the same distribution underlying their attributes, the cross-network relevance is maximized.

Next, we can generalize the naive definition of cross-network relevance to measure the relevance between the re-sampled source network $\overline{\mathcal{G}}^0$ parameterized by the node distribution $\boldsymbol{\beta}$ and the target network $\mathcal{G}$ in our problem. Instead of averaging over all nodes in the source network, we need to compute the expected value based on the sampling distribution $\boldsymbol{\beta}$. Consider a node U sampled from the node set $\mathcal{V}^0$ according to the distribution $\boldsymbol{\beta}$ in the re-sampled source network. Its average relevance to the nodes in the target network is defined as follows:

$$\overline{R}(U, \mathcal{G}) = \frac{1}{m} \sum_{s=1}^{m} S(U, v_s). \tag{7}$$

Since U is a random variable from $\mathcal{V}^0$, the function $R(U, \mathcal{G})$ is also a random variable, for which we can compute an expected value. This provides a measure of the cross-network relevance between the re-sampled source network and the target network. Thus, we have:

$$\begin{aligned} \mathbb{E}_{V \sim \boldsymbol{\beta}} \overline{R}(U, \mathcal{G}) &= \mathbb{E}_{V \sim \boldsymbol{\beta}} \frac{1}{m} \sum_{s=1}^{m} S(U, v_s) \\ &= \frac{1}{m} \sum_{s=1}^{m} \mathbb{E}_{V \sim \boldsymbol{\beta}} S(U, v_s) = \frac{1}{m} \sum_{s=1}^{m} \sum_{i=1}^{n} \beta_i S(v_i^0, v_s) \\ &= \frac{1}{m} \sum_{s=1}^{m} \sum_{i=1}^{n} \beta_i S(v_i^0, v_s). \end{aligned} \tag{8}$$

When $\boldsymbol{\beta}$ is uniformly distributed on the source network, $\beta_i = \frac{1}{n}$, the above equation reduces to afore-mentioned naive definition of Eq. (6). Then, we define the cross-network relevance between the re-sampled source and the target networks as follows:

$$\mathrm{Rel}(\overline{\mathcal{G}}^0) = \mathbb{E}_{V \sim \boldsymbol{\beta}} \overline{R}(V, \mathcal{G}) = \boldsymbol{\beta}' \mathbf{u}, \tag{9}$$

where $'$ is the transpose operator, and $\mathbf{u}$ is a $n \times 1$ vector as

$$\mathbf{u} = \frac{1}{m} \left[\sum_{s=1}^{m} S(v_1^0, v_s), \sum_{s=1}^{m} S(v_2^0, v_s), \ldots, \sum_{s=1}^{m} S(v_n^0, v_s) \right]^T. \tag{10}$$

It is noteworthy that we measure the cross-network relevance based on the node attributes instead of the link attributes. This is essential, because the target network is typically nascent, and sufficient links may not be available for robustly creating such a measure.

The maximization of this cross-network relevance ensures the determination of a distribution $\boldsymbol{\beta}$, which ensures that the re-sampled source network is as relevant as possible for the transfer process. However, it does not guarantee the *richness* of the network structure, which ensures that a sufficient amount of network structure is available for the transfer learning process. Therefore, we need to create an additional component for the objective function for optimizing $\boldsymbol{\beta}$, which measures link richness. The optimization of this combined measure provides a way to tradeoff between the cross-network relevance and link richness.

5.1.2 Link Richness

In this section, we will discuss the contribution of the link richness to the objective function for optimizing the sampling weights $\boldsymbol{\beta}$. Consider two nodes U, W that are independently sampled from $\mathcal{V}^0$ according to the distribution $\boldsymbol{\beta}$ in the re-sampling process. We can compute the probability that they sample a link $(v_i^0, v_j^0) \in \mathcal{E}^0$ of the original source network as $\Pr((U, W) = (v_i^0, v_j^0)) \propto \beta_i \cdot \beta_j$ in Eq. (5). Then, we can sum up all the sampling probabilities of the links in

the source network to measure the proportion of the preserved links:

$$\sum_{i=1}^{n}\left(\frac{1}{k_i}\sum_{j\in\mathcal{N}_i}\beta_i\cdot\beta_j\right). \tag{11}$$

Here, $\mathcal{N}_i$ represents the set of neighbors of the node v_i^0 in the source network, and $k_i=|\mathcal{N}_i|$ is the node degree. For each node v_i, we measure the average sampling probability over all the links incident with it, and then sum over all the nodes in the network. The damping factor $\frac{1}{k_i}$ ensures that densely linked nodes are not over-sampled excessively as compared with the sparsely linked nodes. Maximizing the above results in a rich network, which preserves as much link structure as possible.

We also need to regularize the sampling weight β_i for each node to prevent over-sampling of some nodes in the source networks. The following represents the regularization terms for the link richness optimization problem on the sampling weights $\boldsymbol{\beta}$:

$$\frac{1}{2}\sum_{i=1}^{n}\left(1+\sum_{j\in\mathcal{N}_i}\frac{1}{k_j}\right)\cdot\beta_i^2. \tag{12}$$

This equation suggests that the sampling weight β_i of a node v_i^0 should be penalized by an extra factor $\frac{1}{k_j}$ when it is linked to a neighbor node v_j^0. In other words, two neighboring nodes will compete for the distribution of their sampling weights, and the node with dense links should be penalized to a greater degree to avoid being over-sampled. This guarantees that a sparsely linked node can still sufficiently be sampled, so as to not lose the overall structural information in the network.

Combining the richness objective function of Eq. (11) with the regularization of Eq. (12), we maximize the following regularized link richness expression:

$$\begin{aligned}\text{LinkRich}\left(\overline{\mathcal{G}}^0\right)=&\sum_{i=1}^{n}\left(\frac{1}{k_i}\sum_{j\in\mathcal{N}_i}\beta_i\cdot\beta_j\right)\\&-\frac{1}{2}\sum_{i=1}^{n}\left(1+\sum_{j\in\mathcal{N}_i}\frac{1}{k_j}\right)\cdot\beta_i^2=\boldsymbol{\beta}'\cdot A\cdot\boldsymbol{\beta}.\end{aligned} \tag{13}$$

Here, A is a $n\times n$ matrix, with $A_{ii}=-\frac{1}{2}(1+\sum_{j\in\mathcal{N}_i}\frac{1}{k_j})$, $A_{ij}=\frac{1}{k_i}$ for $j\in\mathcal{N}_i$, and $A_{ij}=0$ otherwise.

The impact of link richness can be explored in terms of the derivative of the objective function that quantifies it. The derivative of this link richness function with respect to a node-specific sampling weight β_i is as follows:

$$\partial_i=\left(\frac{1}{k_i}\sum_{j\in\mathcal{N}_i}\beta_j-\beta_i\right)+\sum_{j\in\mathcal{N}_i}\frac{1}{k_j}(\beta_j-\beta_i). \tag{14}$$

If the neighbors of v_i^0 are strongly relevant to the target network with a greater average sampling weight, the first term will become positive and force the sampling weight β_i to increase, so as to preserve its incident links. In the second term, for each neighbor v_j^0 of the node v_i^0, if it is relevant to the target network with a greater β_j, it will also tend to increase β_i to preserve the incident link. Moreover, when v_j^0 is sparsely linked with a small node degree k_j, β_i will increase in order to preserve this link that is more informative to v_j^0 as compared with the links incident with the other densely linked nodes in the network. Fig. 3 illustrates an example. The node v_1^0 has a set of neighboring nodes $\{v_2^0,v_3^0,v_4^0,v_5^0\}$ that have greater sampling weights on the average. In order to preserve the links between them and v_1^0, the sampling weight of the central node v_1^0 tends to be increased. On the other hand, among these neighboring nodes, v_5^0 has only one incident link, which is important to preserve. As indicated by the second term, the sampling weight of the central node v_1^0 should be increased more to preserve this link.

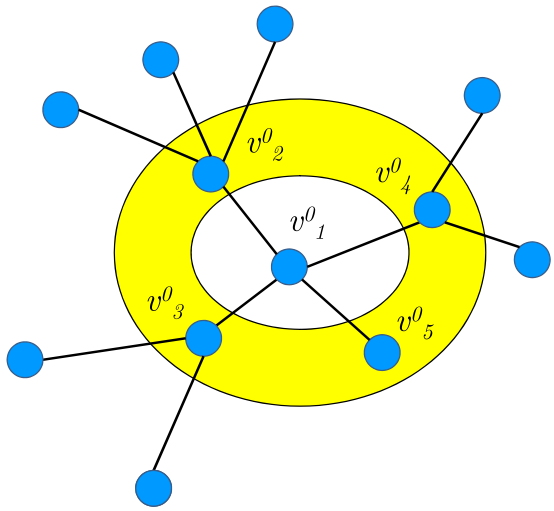


Fig. 3. An example of a central node v_1^0, and its neighborhood. The re-sampling weight of the central node will be increased by greater average re-sampling weight of its neighbors.

It is evident that the sampling weight of a singleton node with no incident link will be zero, because the exclusion of such a node does not lose link information. For a singleton node v_i^0, the derivative in Eq. (14) becomes $\partial_i=-\beta_i$. This will decrease β_i until it reaches zero.

An information Theoretic Perspective of Link Richness. We can find that the derivative in Eq. (14) disappears when all the nodes in the source network are uniformly sampled with an equal sampling weight, except for the singleton nodes whose weights are set to zero instead. From the information theoretic perspective, the uniform distribution over the nodes gives rise to the maximum information we can obtain about the link structure for the network. Excluding the singleton nodes does not lose any information about link structure since it contains no link information. This will provide a compact support over the non-singleton nodes in the network. In other words, according to the link richness defined above, all non-singleton nodes tend to be sampled *uniformly*, which reduces the loss of many important global link properties, such as the power-law distribution of node degrees in many real-world networks, invariant network diameters, and the (dis)connectivity between the nodes in networks. In this sense, our definition of link richness provides a reasonable measurement of link information, not only locally but also globally.

5.1.3 Putting It All Together: Combining Cross-Network Relevance and Link Richness

The optimal sampling distribution $\boldsymbol{\beta}$ can be obtained by maximizing the cross-network relevance $\mathrm{Rel}(\overline{\mathcal{G}}^0, \mathcal{G})$, as well as the link richness $\mathrm{LinkRich}(\overline{\mathcal{G}}^0)$ of the re-sampled source network simultaneously. Therefore, we create a combined objective function for optimization as follows:

$$\begin{aligned}\boldsymbol{\beta}^\star &= \operatorname*{argmax}_{\boldsymbol{\beta}} \mathrm{Rel}(\overline{\mathcal{G}}^0, \mathcal{G}) + \lambda \cdot \mathrm{LinkRich}(\overline{\mathcal{G}}^0) \\ &= \operatorname*{argmax}_{\boldsymbol{\beta}} \boldsymbol{\beta}'\mathbf{u} + \lambda \cdot \boldsymbol{\beta}' A \boldsymbol{\beta} \\ &s.t., \sum_{i=1}^{n} \beta_i = 1, \beta_i \geq 0.\end{aligned} \tag{15}$$

Here, λ is the parameter that weighs the relative importance of the bias-correction and link richness criteria in the optimization process. The projected gradient method [27] can be applied to optimize the sampling weights $\boldsymbol{\beta}$ iteratively as follows:

$$\begin{aligned}\boldsymbol{\beta} &\leftarrow \prod_{\mathcal{S}}\left[\boldsymbol{\beta} + \delta \nabla_{\boldsymbol{\beta}}\{\boldsymbol{\beta}'\mathbf{u} + \lambda \cdot \boldsymbol{\beta}^T A \boldsymbol{\beta}\}\right] \\ &= \prod_{\mathcal{S}}[\boldsymbol{\beta} + \delta\{\mathbf{u} + 2\lambda A\boldsymbol{\beta}\}].\end{aligned} \tag{16}$$

Here, δ is the step size along the gradient direction in each iteration. The operator $\Pi_{\mathcal{S}}[\cdot]$ is the projection onto the simplex $\mathcal{S}$ defined by the constraint as $\mathcal{S} = \{\boldsymbol{\beta} \in \mathbb{R}^n | \sum_{i=1}^{n} \beta_i = 1, \beta_i \geq 0\}$. We use an efficient algorithm discussed in [28] in order to determine the projection operator, and determine the optimal solution with iterative application of the projected gradient method. Before we finish this section, we would like to comment on a case that our re-sampling technology reveals when the link transfer may become difficult. When the most relevant nodes in the source network are far apart from each other, e.g., the shortest path connecting them is quite long and the nodes on the path are often irrelevant to the target network, our resampling technology will not be able to preserve the link structure between them if we still want to only retain the most relevant part of the source.

6 Experiments

In this section, we will test the effectiveness of our approach for cross-network link prediction. We will demonstrate the overall effectiveness of the approach, as well as the effectiveness of the bias-correction process.

6.1 Experimental Setup and Data Description

We test the cross-network link transfer algorithms in two different settings.

First, we collect co-authorship networks from the academic publication in four areas—database, data mining, machine learning and information retrieval. It contains the papers published in 20 major conferences with 28,702 authors. Two authors are linked in the network if they collaborate on a paper. This totally forms 66,832 coauthor links, and each author is linked with 2.3 coauthors on average. The attributes of the authors in the network are represented by the 13,214 keywords that are extracted from the title of their publications. Then Term Frequency and Inverse Document Frequency (TFIDF) features are computed as the attribute vector for each author.

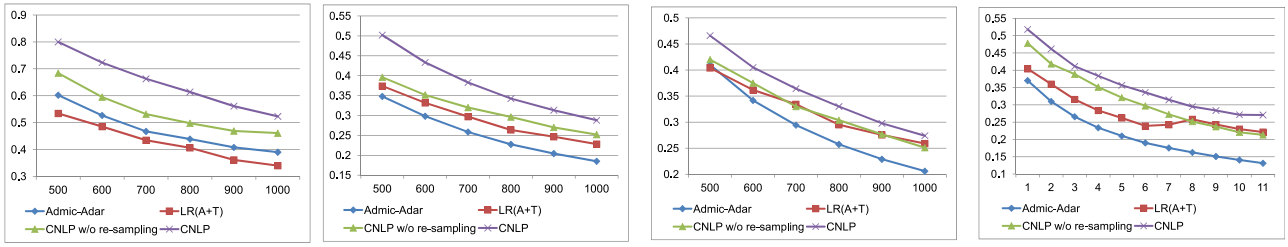
Specifically, we combine the publications in three of these four areas to construct the source network, and the publications in the remaining area are used to construct the target network. Thus, we can use this approach to construct four different data sets, by varying the target network. For the source network, all the publications are used to extract the links and attributes in the network. In the target network, we retain the links and attributes from 20 percent of the publications in order to create the nascent target network. Our goal is to predict the remaining co-authorship links. This is a challenging link prediction task, because the link structure of the target network is very sparse.

Second, we also transfer link structure across the co-author networks extracted from two different data sets—the Cora research paper data set and the above DBLP data set. The Cora research paper data set is derived from the original Cora Research Paper data set at http://www.cs.umass.edu/ mccallum/code-data.html. It has 24,961 authors of 19,396 research papers. The goal is to use the coauthor link information on Cora data set to predict the missing links on the same DBLP data set as above. Cora and DBLP are independently established and maintained, and the co-author link graph underlying two data set can have different coverage of authors and research areas. Actually, in addition to some closer research areas like machine learning in artificial intelligence, Cora data set contains the authors from the areas such as operating systems, security, networking, hardware and architecture. These research areas are quite different from the four areas in the above DBLP data set. This necessitates the rectification of cross-network bias between these two data sets, so that the link model built upon the Cora data set can well capture the link structure for the DBLP data set.

6.2 Baseline Algorithms

We compare the proposed link prediction algorithm with the following benchmark algorithms:

- Adamic-Adar [11]: It predicts the links between two authors by their common neighbors in the network. The neighboring nodes are weighted by taking the inverse logarithm of their node degrees. The comprehensive study conducted in [7] showed this topological feature about the network link structure was particularly useful for link prediction. Therefore, we adopt it for comparison here as the baseline for link prediction.
- LR(A+T): It combines the attribute (A) similarity and topological features (T) such as the Adamic-Adar between the authors in the network by logistic regression to predict the links in the target network [6]. The logistic regression model is trained on both the source network as well as the current target network with the existing links.
- CNLP without re-sampling: This is the cross-network link prediction model proposed in Section 3, but without re-sampling process proposed in Section 4. It demonstrates the baseline performance of our model



(a) Database (b) Data Mining (c) Machine Learning (d) Information Retrieval

Fig. 4. Top-K precision for the target co-authorship networks (a) Database, (b) Data Mining, (c) Machine Learning and (d) Information Retrieval. For each target network, the other three networks are combined as the source network.

without cross-network bias correction for the link transfer process.

- CNLP: This is the proposed cross-network link prediction model with the re-sampling algorithm. The purpose of using two variations of CNLP is to show the impact of cross-network bias correction on the transfer learning process for link prediction.

The parameters in these algorithms are tuned by a five-fold cross-validation process on the current target network. The link prediction performance is measured based on the top-K precision with K ranging from 500 to 1,000.

6.3 Link Prediction Results

Link prediction between four areas in DBLP. Fig. 4 illustrates the link prediction results for the four target networks. For each target network, we report the top-K precision results from $K = 500$ to $K = 1{,}000$. It is evident that the Adamic-Adar algorithm is not as effective as the other methods, since it only considers the topological features about the network structure. The exception arises in the case of the *Database* network, where it performs better than LR (A+T) that combines the topological features as well as attribute features. This is probably a result of over-fitting to the sparse link structure in the target network in this case. The CNLP without re-sampling outperforms Adamic-Adar and LR(A+T) as it simultaneously explores the target and source network structures for link prediction in the target network. However, its performance is still not the best, and may sometimes become comparable with LR(A+T), as in the case of the *Machine Learning* and *Information Retrieval* networks (c.f. Figs. 4c and 4d). By incorporating the re-sampling algorithm to correct the cross-network bias, CNLP achieves the best performance in link prediction. It avoids sampling the inconsistent link structure in the target network, and improves the quality of link transfer process in our problem. In the following section, we will examine the re-sampling results at a more detailed level, and understand why CNLP can perform better compared to other algorithms.

Link prediction between Cora and DBLP. Fig. 5 compares the Top-K precisions of the four algorithms for link prediction on the target DBLP network. We can find that without rectifying the cross-network bias, the CNLP w/o re-sampling performs worse than LR(A+T), since the former algorithm has incorporated the coauthor links from irrelevant research areas in the source Cora network. These irrelevant links usually impose bias that makes the link model overwhelmed by the authors with the profiles and expertise distinct from the DBLP network.

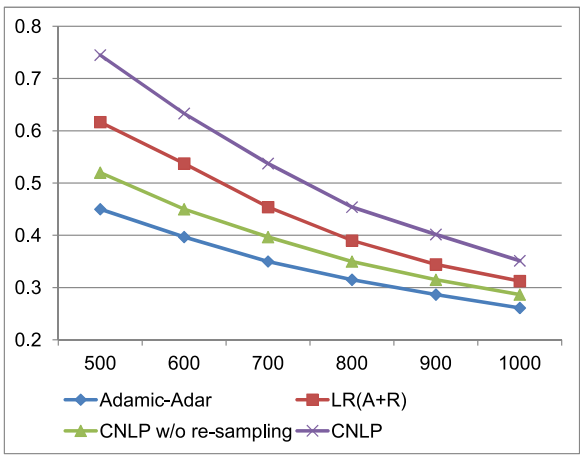


Fig. 5. Top-K precision for predicting the co-authorship links in the target DBLP network.

6.4 Re-Sampling Results

Fig. 6 illustrates the re-sampling results on the source networks for the four target networks. For each target network, the collaboration information in the other three research areas is combined to form the source network. In this figure, we illustrate the average re-sampling weight on each link with respect to the different research areas. We can see that the re-sampling results reflect the relevance between these research areas well. For example, in Fig. 6a, the *Data Mining* area is the most relevant to the *Database* area, as they usually share many common research topics evident from the top-ranked keywords in Table 2. Moreover, in Table 3, we give the top-10 keywords

TABLE 1
The Conferences in Four Different Research Areas

Database	Data Mining	Machine Learning	Information Retrieval
ICDE	KDD	IJCAI	SIGIR
VLDB	PAKDD	AAAI	CIKM
SIGMOD	ICDM	ICML	WWW
PODS	PKDD	CVPR	ECIR
EDBT	SDM	ECML	WSDM

The publications that are used to extract the co-authorship links and author attributes are obtained from these 20 conferences.

TABLE 2
The Conferences in Four Different Research Areas

Database	Data Mining	Machine Learning	Information Retrieval
data	data	learning	retrieval
database	mining	knowledge	information
query	clustering	reasoning	web
queries	efficient	system	search
databases	learning	approach	text
system	databases	systems	query
xml	patterns	logic	document
systems	large	model	model
efficient	frequent	search	system
processing	classification	data	classification

The publications that are used to extract the co-authorship links and author attributes are obtained from these 20 conferences.

associated with the top-100 authors re-sampled in the source network. The keywords that appear in the top-10 keywords of the corresponding target network as in Table 2 are highlighted in bold. This also shows that the re-sampling algorithm corrects cross-network bias. This explains the better generalization performance for the CNLP algorithm in the target network.

6.5 Computational Efficiency

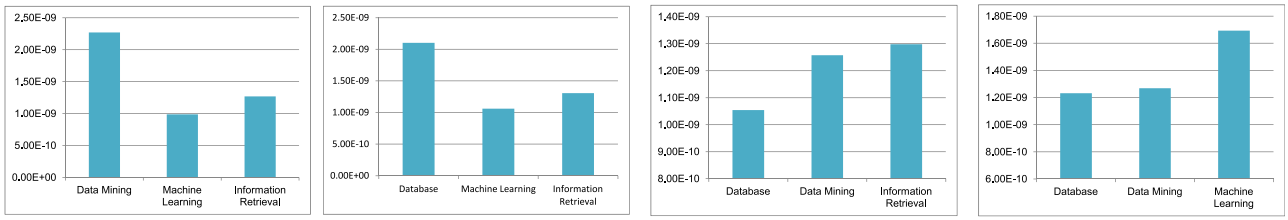
Finally, we report the running time of our algorithm on the four target networks. The experiments were conducted on an Intel(R) Xeon(R) 2.40 GHz CPU processor with 8 GB physical memory and Linux system. The algorithms required about 38.36 seconds to build the link model and 57.70 seconds to re-sample the source network. Once the link model was built, the link between the authors could be predicted in 2.70×10^{-5} milliseconds. In comparison, Adamic-Adar and LR(A+T) took 0.86×10^{-5} milliseconds and 1.38×10^{-5} milliseconds respectively to predict the link between the authors. Thus, it is shown that while our approach provides more accurate link prediction, its computation cost is also comparable as the other algorithms.

6.6 Parameter Sensitivity

We also test the parameter sensitivity of the CNLP model here. Table 15 reports the change of the AUC measures with different λ that trades off between the cross-network relevance and link richness as in Eq. (15). Here AUC is a measure that computes the area under the ROC curve, which measures the probability that a positive link is put at a higher rank than a negative link. Compared with the Top-K precision we used in Fig. 4, AUC provides us a quantity summarizing the overall precision of the resulting rank list of predicted links. For the cross-network link prediction between four areas, we report the average AUC with each different area as target network and the other three as source network. For the link transfer from Cora to DBLP, we directly report the AUC result.

From the table, we can see that the model performance depends on a balanced consideration of cross-network relevance and link richness. To one end, when λ is set to a smaller value, only a smaller part of source network is re-sampled, and a lot of rich link information is lost accordingly. Extremely, only few nodes most relevant to the target network will be retained. In this case, the link prediction model is built upon a very sparse re-sampled source network, which is inadequate to predict the links in the target network.

To the other end, when λ is set to a larger value, almost the whole source network is retained without selectivity, no matter if a particular part of network is relevant or not to



(a) Database (b) Data Mining (c) Machine Learning (d) Information Retrieval

Fig. 6. For the target network (a) Database, (b) Data Mining, (c) Machine Learning and (d) Information Retrieval, the table shows the average re-sampling weights for the links in the source network.

TABLE 3
The Top-10 Keywords Associated with the Top-100 Authors Re-Sampled in the Source Network

Four areas in DBLP				From Cora to DBLP
Database	Data Mining	Machine Learning	Information Retrieval	
data	**data**	**data**	data	models
mining	**efficient**	web	learning	planning
database	queries	information	**web**	analysis
learning	database	mining	database	multi
query	query	search	**model**	data
databases	**learning**	retrieval	**system**	algorithm
efficient	**databases**	**learning**	**query**	classification
queries	**mining**	databases	**information**	constraint
clustering	xml	query	systems	efficient
system	web	**model**	**text**	information

The keywords in bold appear in the corresponding top-10 keywords associated with each target network as in Table 2. It is evident that the re-sampling process captures the information in the source networks that is relevant to the target network.

TABLE 4
Change of AUC Measures with Different λ that Trades Off between the Cross-Network Relevance and Link Richness As in Eq. (15)

λ	Between Four Areas	Cora to DBLP
0.1	0.604	0.677
0.2	0.725	0.742
0.5	0.803	0.824
1.0	0.867	0.895
2.0	0.915	0.936
4.0	0.903	0.927
10.0	0.884	0.914
15.0	0.872	0.878
20.0	0.870	0.869

the target network. In this case, the performance of the link prediction model is also affected adversely by the irrelevant source network structure. Therefore, the model parameter should be properly set to ensure a balanced consideration of cross-network relevance and link richness.

7 Conclusions

In this paper, we introduce the problem of cross-network link prediction. The idea is to capture the rich linkage structure in existing networks in order to predict links in nascent target networks. A robust link transfer model is proposed for efficient link knowledge transfer between the networks. The cross-network bias in the problem is corrected by resampling the source network to avoid model over-fitting. We present experimental results on real networks in order to demonstrate the advantages of our approach over existing methods.

Acknowledgments

Research was sponsored by the Army Research Laboratory and National Science Foundation and was accomplished under Cooperative Agreement Number W911NF-09-2-0053 and Grant IIS-1144111. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. The first author was also in part supported by an IBM Fellowship.

References

[1] S. F. Adafre and M. Rijke, "Discovering missing links in wikipedia," in *Proc. 3rd Int. Workshop Link Discovery*, 2005, pp. 90–97.
[2] M. Al-Hassan, V. Chaoji, S. Salem, and M. J. Zaki, "Link prediction using supervised learning," in *Proc. Workshop Link Anal., Counter-Terrorism Security*, 2005.
[3] A. Popescul, L. Ungar, S. Lawrence, and D. Pennock, "Statistical relational learning for document mining," in *Proc. IEEE Int. Conf. Data Mining*, 2003, pp. 275–282.
[4] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," in *Proc. Advances in Neural Information Process. Syst.*, 2003.
[5] H. Kashima and N. Abe, "A parameterized probabilistic model of evolution for supervised link prediction," in *Proc. 6th Int. Conf. Data Mining*, 2006, pp. 340–349.
[6] C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction," in *Proc. Seventh IEEE Int. Conf. Data Mining*, 2007, pp. 322–331.
[7] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proc. 12th Int. Conf. Inf. Knowl. Manag.*, 2004, pp. 556–559.
[8] J. R. Doppa, J. Yu, P. Tadepalli, and L. Getoor, "Chance constrained programs for link prediction," in *Proc. NIPS Workshop Analyzing Netw. Learning Graphs*, 2009.
[9] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Shölkopf, "Correcting sample selection bias by unlabeled data," in *Proc. Adv. Neural Inform. Process. Syst.*, 2006, pp. 601–608.
[10] J. Doppa, J. Yu, P. Tadepalli, and L. Getoor, "Link mining: A survey," *SIGKDD Explorations*, vol. 7, pp. 3–12, 2005.
[11] L. Adamic and E. Adar, "Friends and neighbors on the web," *Social Netw.*, vol. 25, pp. 211–230, 2001.
[12] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Phys. Rev. Lett.*, vol. 64, p. 025102, 2001.
[13] M. Bilgic, G. Namata, and L. Getoor, "Combining collective classification and link prediction," in *Proc. Workshop Mining Graphs Complex Struct. (IEEE Int. Conf. Data Mining)*, 2007, pp. 381–386.
[14] L. Getoor, N. Friedman, D. Koller, and B. Taskar, "Learning probabilistic models of relational structure," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 170–177.
[15] L. Getoor, N. Friedman, D. Koller, and B. Taskar, "Learning probabilistic models of link structure," *J. Mach. Learn. Res.*, vol. 3, pp. 679–707, 2002.
[16] O. Hassanzadeh, A. Kementsietsidis, L. Lim, R. J. Miller, and M. Wang, "A framework for semantic link discovery over relational data," in *Proc. 18th ACM Conf. Inform. Knowl. Manag.*, 2009, pp. 1027–1036.
[17] B. Taskar, P. Abbeel, and D. Kollerk, "Discriminative probabilistic models for relational data," in *Proc. 18th Conf. Uncertainty Artif. Intell.*, 2002, pp. 485–492.
[18] R. Raina, A. Ng, and D. Koller, "Constructing informative priors using transfer learning," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 713–720.
[19] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[20] J. Tang, T. Lou, and J. M. Kleinberg, "Inferring social ties across heterogenous networks," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 743–752.
[21] J. Ye, H. Cheng, Z. Zhu, and M. Chen, "Predicting positive and negative links in signed social networks by transfer learning," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1477–1488.
[22] A. Narayanan, E. Shi, and B. I. P. Rubinstein, "Link prediction by de-anonymization: How we won the kaggle social network challenge," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 1825–1834.
[23] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 243–252.
[24] G. Schwartz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 1979.
[25] R.-H. Li, J. X. Yu, and J. Liu, "Link prediction: The power of maximal entropy random walk," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manag.*, 2011, pp. 1147–1156.
[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[27] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Oper. Res. Lett.*, vol. 31, no. 3, pp. 167–175, May 2003.
[28] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proc. Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 272–279.

Guo-Jun Qi received the PhD degree from the University of Illinois at Urbana-Champaign, in December 2013. His research interests include pattern recognition, machine learning, computer vision, multimedia, and data mining. He received twice IBM PhD fellowships, and Microsoft fellowship. He is the recipient of the Best Paper Award at the 15th ACM International Conference on Multimedia, Augsburg, Germany, 2007. He is currently a faculty member with the Department of Electrical Engineering and Computer Science at the University of Central Florida, and has served as program committee member and reviewer for many academic conferences and journals in the fields of pattern recognition, machine learning, data mining, computer vision, and multimedia.

Charu C. Aggarwal received the BS degree from IIT Kanpur in 1993 and the PhD degree from Massachusetts Institute of Technology in 1996. He is a research scientist at the IBM T.J. Watson Research Center in Yorktown Heights, New York. He has since worked in the field of performance analysis, databases, and data mining. He has published more than 155 papers in refereed conferences and journals, and has been granted over 50 patents. He has served on the program committees of most major database/data mining conferences, and served as program vice-chairs of the SIAM Conference on Data Mining, 2007, the IEEE ICDM Conference, 2007, the WWW Conference 2009, and the IEEE ICDM Conference, 2009. He served as an associate editor of the *IEEE Transactions on Knowledge and Data Engineering Journal* from 2004 to 2008. He is an associate editor of the *ACM TKDD Journal*, an action editor of the *Data Mining and Knowledge Discovery Journal*, an associate editor of the *ACM SIGKDD Explorations*, and an associate editor of the *Knowledge and Information Systems Journal.* He is a fellow of the IEEE for "contributions to knowledge discovery and data mining technique", and a life member of the ACM.

Thomas S. Huang received the ScD from MIT in 1963. He is a William L. Everitt Distinguished professor in the Department of Electrical and Computer Engineering at the University of Ilinois at Urbana-Champaign. Since 2001, he has also been the member of National Academy of Engineering. Before he joined the University of Illinois, he was a professor at Purdue University from 1973 to 1980, and an assistant and then an associate professor at MIT from 1963 to 1973, both in the Department of Electrical Engineering. His professional interests are computer vision, image processing, pattern recognition, and multimodal signal processing. He is a life fellow of the IEEE.