# Link Prediction across Networks by Biased Cross-Network Sampling

Guo-Jun Qi[1], Charu C. Aggarwal[2], Thomas Huang[1]

[1]*Department of Electrical and Computer Engineering*
*University of Illinois at Urbana-Champaign*
{qi4, t-huang1}@illinois.edu

[2]*IBM T.J. Watson Research Center*
charu@us.ibm.com

***Abstract*— The problem of link inference has been widely studied in a variety of social networking scenarios. In this problem, we wish to predict future links in a growing network with the use of the existing network structure. However, most of the existing methods work well only if a significant number of links are already available in the network for the inference process. In many scenarios, the existing network may be too sparse, and may have too few links to enable meaningful learning mechanisms. This paucity of linkage information can be challenging for the link inference problem. However, in many cases, other (more densely linked) networks may be available which show similar linkage structure in terms of underlying attribute information in the nodes. The linkage information in the existing networks can be used *in conjunction with the node attribute information* in both networks in order to make meaningful link recommendations. Thus, this paper introduces the use of transfer learning methods for performing cross-network link inference. We present experimental results illustrating the effectiveness of the approach.**

## I. Introduction

The problem of link inference is that of predicting links between nodes in a social network based on its current structure and the content in the nodes [1], [2], [3], [4], [5], [6], [7], [8]. Most of the known techniques heavily use the local structure in the current network in order to perform the link inference. Specifically, two nodes are more likely to be linked, if they are structurally connected through already existing nodes. For example, in the case of social networks such as *Facebook*, link recommendations are made between two nodes, when the two nodes are already connected indirectly through many common friends.

It has been shown in [7] that the structural information in a network is an extremely powerful and reliable source of information for the purpose of link inference. Some examples of such structural information could include the number of common neighbors, or other local neighborhood criteria. However, such an approach is useful only when a sufficient amount of structure is already available in the network for performing the inference. For networks in a stage of infancy, such an approach is not very useful, because a given pair of nodes may not have a lot of common neighbors (or connected by a short path), even when they are closely related to one another. Therefore, the traditional methods for link inference fail in the case of sparse networks because of the paucity of available structural information. The link inference problem is particularly important for the case of sparse (or new) networks which still have significant room to grow, and the basic structure of the networks is not known to a large degree. *Therefore, the existing methods for link inference are particularly challenged in scenarios which are the most important for the link inference problem.*

In this paper, we will study the cross-network link prediction problem, which attempts to leverage the existing link information in a mature source network (eg. *Facebook*) to predict the links in a relatively new network (eg. *Google+*). The link inference problem can also be treated as a classification problem in which *derived attributes between node pairs*, such as structural similarity or attribute similarity can be used as training data for cases in which existence of links is known. This can then be used for predicting links between node pairs in the target network where their (future) existence is unknown. The connections between the link inference problem and the classification problem point to a natural approach in which the structural and attribute information in the training network is used in order to enhance the link inference problem in the target network. This is related to the problem of *transfer learning* [9], which is commonly used in machine learning scenarios in which paucity of data in one domain is used to enhance the learning process in another domain.

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

- If two nodes containing the keywords { *John Smith*, *Ohio* } and { *Kevin Jordan*, *Cincinnati* } are connected together in one network, then this can also provide a hint for the target network. However, it is not necessary that the correspondingly named nodes in the target network correspond to the same people. However, if the node containing { *Kevin Jordan*, *Cincinnati* } is connected to similar nodes in both networks, this can provide a powerful hint about the linkage behavior. This is of course a case where the link inference is related to the

problem of exact identity-based node matching, though much more ambiguous hints are possible in which there may not be exact matching of the nodes. Some examples of such cases follow.

- 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 which is highly related to linkage behavior. Therefore, the transfer process needs to *learn* the content which 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 { *UIUC machine learning* } may be highly linked with nodes containing { *UIUC data mining* }, though it may not be as closely related to nodes containing the keywords { *UIUC English Literature* }. 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 { *UIUC machine learning* } and { *UIUC 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.

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. We will formalize the cross-network link prediction problem and present the main ideas and challenges in Section 2. A link model is built on the source network in Section 3. In section 4, 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 3. We present experimental results in Section 5 to demonstrate the effectiveness of our approach. The conclusions and summary are presented in Section 6.

### A. Related Work and Contributions

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 [11], [7], [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 [13], [14], [15], [16], [3], [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. Moreover, recently 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.

## II. Problem Formulation

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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ which is assumed to be nascent and sparse. The node sets in the source and target networks are denoted by $\mathcal{V}^0 = \{v_1^0, v_2^0, \cdots, v_m^0\}$ and $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ 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 which 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 which 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 which 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 which 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 2.1 (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 which 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 $\overline{x}$.

### A. 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 which 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.

## III. 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 $i$th and $j$th 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 which 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 which 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 $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ 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. Specifically, the content vectors $\mathbf{x}_i^0$ for each node $v_i^0$ in $\mathcal{G}^0$ and $\mathbf{x}_i$ 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) [21]. 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, 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 (CN) 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]. 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 which 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 which is transferred from the source to the target network.

In the following, we will design an objective function which 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 $\mathcal{G}$:

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

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 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 which 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:

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

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 aforementioned 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 $\mathbf{W}$:

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

The last term imposes a regularizer for better generalization performance, and $\eta$ and $\gamma$ 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 unconstraint optimization solver such as conjugate gradient method [22].

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

## IV. 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.

Figure 1 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 Figure 1 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.

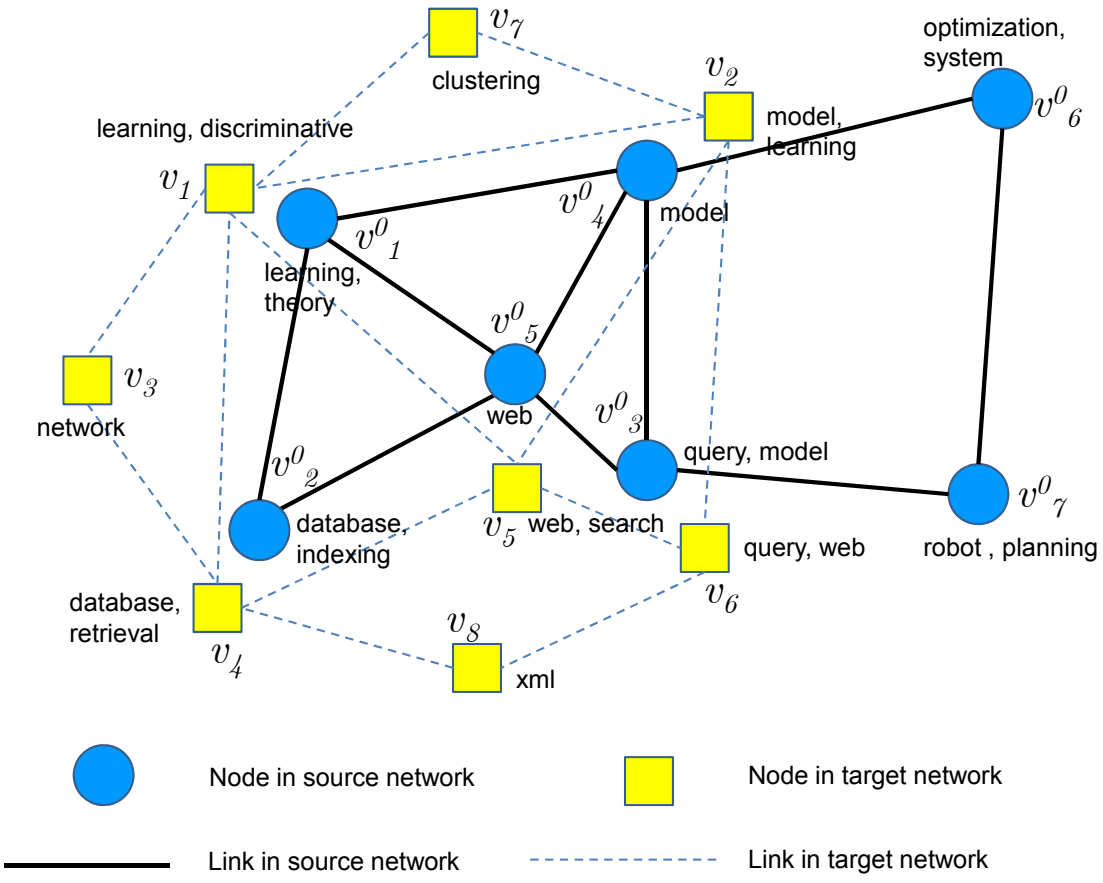



Fig. 1. An example of the bias in attribute information associated with source and target networks

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.

### A. Re-sampling the Source Network

In the re-sampled source network, each node is sampled in iid fashion according to a weighting distribution $\boldsymbol{\beta} = \{\beta_1, \beta_2, \cdots, \beta_n\}$ 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 $\bar{\mathcal{G}}^0 = \{\bar{\mathcal{V}}^0, \bar{\mathcal{E}}^0, \boldsymbol{\beta}\}$ 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 $\bar{\mathcal{G}}^0$ is a random variable which takes on values from $\bar{\mathcal{V}}^0$ with the probability that $\Pr(U = v_i^0) = \beta_i$ for $i = 1, 2, \cdots, n$.*

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:

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

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 which 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 subsections, and the optimal determination of the distribution $\boldsymbol{\beta}$ on this basis.

*1) Cross-Network Relevance:* The relevance $R\left(\mathcal{G}^0, \mathcal{G}\right)$ 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:

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

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 $\bar{\mathcal{G}}^0$ 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:

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

Since $U$ is a random variable from $\mathcal{V}^0$, the function $R\left(U, \mathcal{G}\right)$ 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:

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

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:

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

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

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

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.

*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 $\Pr\left((U, W) = (v_i^0, v_j^0)\right) \propto \beta_i \cdot \beta_j$ 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:

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

Here, $\mathcal{N}_i$ represents the set of neighbors of the node $v_i^0$ in the source network, and $k_i = |N_i|$ 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 $\beta_i$ 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 $\beta_i$ of a node $v_i^0$ should be penalized by an extra factor $\frac{1}{k_j}$ when it is linked

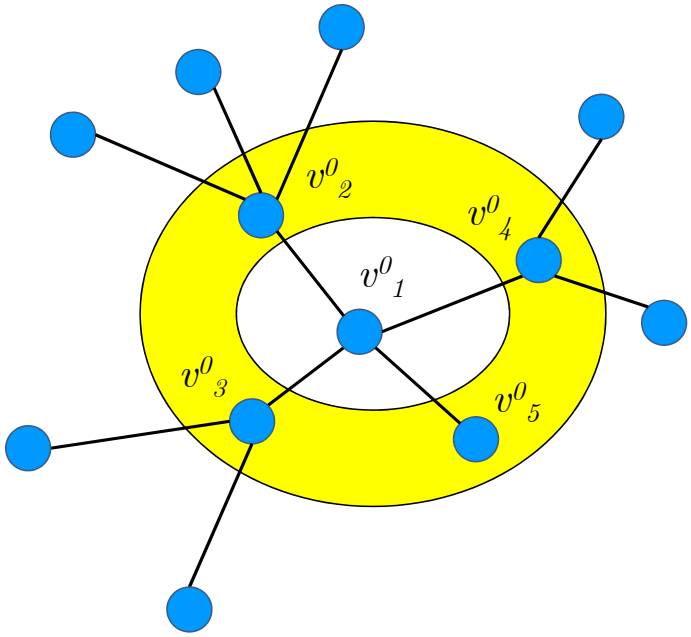



Fig. 2. An example of a central node $v_1^0$, and its neighborhood. The re-sampling weight of the central node will be increased by greater average re-sampling weight of its neighbors.

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 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:

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

Here, $A$ is a $n\times n$ matrix, with $A_{ii}=-\frac{1}{2}\left(1+\sum_{j\in\mathcal{N}_i}\frac{1}{k_j}\right)$, $A_{ij}=\frac{1}{k_i}$ 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 which quantifies it. The derivative of this link richness function with respect to a node-specific sampling weight $\beta_i$ 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 $\beta_i$ 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 $\beta_j$, it will also tend to increase $\beta_i$ to preserve the incident link. Moreover, when $v_j^0$ is sparsely linked with a small node degree $k_j$, $\beta_i$ will increase in order to preserve this link which is more informative to $v_j^0$ as compared with the links incident with the other densely linked nodes in the network. Figure 2 illustrates an example. The node $v_1^0$ has a set of neighboring nodes $\{v_2^0, v_3^0, v_4^0, v_5^0\}$ which 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.

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 $\beta_i$ until it reaches zero.

*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 $\text{Rel}(\bar{\mathcal{G}}^0, \mathcal{G})$, as well as the link richness $\text{LinkRich}(\bar{\mathcal{G}}^0)$ of the re-sampled source network simultaneously. Therefore, we create a combined objective function for optimization as

follows:

$$\begin{aligned}\beta^\star &= \arg\max_{\boldsymbol{\beta}} \text{Rel}\left(\bar{\mathcal{G}}^0, \mathcal{G}\right) + \lambda \cdot \text{LinkRich}\left(\bar{\mathcal{G}}^0\right) \\ &= \arg\max_{\boldsymbol{\beta}} \boldsymbol{\beta}'\mathbf{u} + \lambda \cdot \boldsymbol{\beta}' A \boldsymbol{\beta} \\ &s.t., \sum_{i=1}^{n} \beta_i = 1, \beta_i \geq 0\end{aligned} \tag{15}$$

Here, $\lambda$ is the parameter which weighs the relative importance of the bias-correction and link richness criteria in the optimization process. The projected gradient method [23] 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, $\delta$ is the step size along the gradient direction in each iteration. The operator $\prod_{\mathcal{S}}[\cdot]$ 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 [24] in order to determine the projection operator, and determine the optimal solution with iterative application of the projected gradient method.

## V. 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.

### A. Experimental Setup and Data Description

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 twenty 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 which are extracted from the title of their publications. Then TFIDF (Term Frequency and Inverse Document Frequency) 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 $20\%$ of the publications in order to create the nascent target network. Our goal is to predict remaining the co-authorship links. This is a challenging link prediction task, because the link structure of the target network is very sparse.

### B. 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 (LR) 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 (CNLP) model proposed in Section 3, but without re-sampling process proposed in Section 4. It demonstrates the baseline performance of our model 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 $5$-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 $1,000$.

### C. Link Prediction Results

Figure 3 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) which combines the topological features as well as attribute features. This is probably a result of overfitting 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*

TABLE I

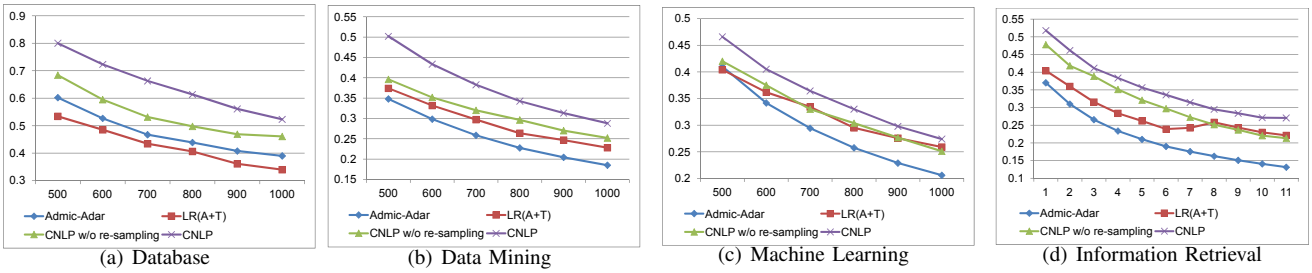
THE CONFERENCES IN FOUR DIFFERENT RESEARCH AREAS. THE PUBLICATIONS WHICH ARE USED TO EXTRACT THE CO-AUTHORSHIP LINKS AND AUTHOR ATTRIBUTES ARE OBTAINED FROM THESE 20 CONFERENCES.

| 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 |

TABLE II

THE CONFERENCES IN FOUR DIFFERENT RESEARCH AREAS. THE PUBLICATIONS WHICH ARE USED TO EXTRACT THE CO-AUTHORSHIP LINKS AND AUTHOR ATTRIBUTES ARE OBTAINED FROM THESE 20 CONFERENCES.

| 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 |



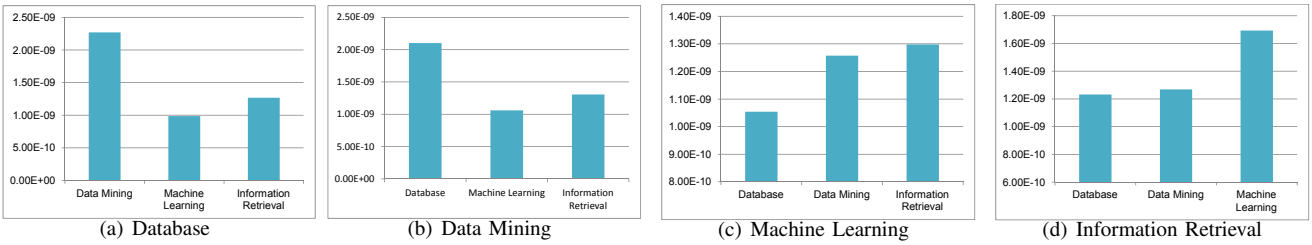

(a) Database (b) Data Mining (c) Machine Learning (d) Information Retrieval

Fig. 3. 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.

networks (c.f. Figure 3(c) and 3(d)). 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 subsection, we will examine the re-sampling results at a more detailed level, and understand why CNLP can perform better compared to other algorithms.

### *D. Re-sampling Results*

Figure 4 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 Figure 4(a), 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 II. Moreover, in Table III, we give the top-10 keywords 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 II are highlighted in bold. This also shows that the re-sampling algorithm corrects cross-network bias. This




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

Fig. 4. 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 III

THE TOP-10 KEYWORDS ASSOCIATED WITH THE TOP-100 AUTHORS RE-SAMPLED IN THE SOURCE NETWORK. THE KEYWORDS IN BOLD APPEAR IN THE CORRESPONDING TOP-10 KEYWORDS ASSOCIATED WITH EACH TARGET NETWORK AS IN TABLE II. IT IS EVIDENT THAT THE RE-SAMPLING PROCESS CAPTURES THE INFORMATION IN THE SOURCE NETWORKS WHICH IS RELEVANT TO THE TARGET NETWORK.

| Database | Data Mining | Machine Learning | Information Retrieval |
|---|---|---|---|
| **data** | **data** | **data** | data |
| mining | **efficient** | web | learning |
| **database** | queries | information | **web** |
| learning | database | mining | database |
| query | query | search | **model** |
| **databases** | **learning** | retrieval | **system** |
| **efficient** | **databases** | **learning** | **query** |
| **queries** | **mining** | databases | **information** |
| clustering | xml | query | systems |
| **system** | web | **model** | **text** |

explains the better generalization performance for the CNLP algorithm in the target network.

### E. 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) 2.40GHz CPU processor with 8GB 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 $2.70 \times 10^{-5}$ milliseconds. In comparison, Adamic-Adar and LR(A+T) took $0.86 \times 10^{-5}$ milliseconds and $1.38 \times 10^{-5}$ 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.

## VI. 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 re-sampling 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.

## ACKNOWLEDGEMENTS

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. 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. Guo-Jun Qi was also supported by an IBM Fellowship.

## REFERENCES

[1] S. F. Adafre and M. Rijke, "Discovering missing links in wikipedia," in *LINK-KDD*, 2005.
[2] M. Al-Hassan, V. Chaoji, S. Salem, and M. J. Zaki, "Link prediction using supervised learning," in *Workshop on Link Analysis, Counter-terrorism and Security (at SDM)*, 2005.
[3] A. Popescul, L. Ungar, S. Lawrence, and D. Pennock, "Statistical relational learning for document mining," in *ICDM*, 2003.
[4] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," in *NIPS*, 2003.
[5] H. Kashima and N. Abe, "A parameterized probabilistic model of evolution for supervised link prediction," in *ICDM*, 2006.
[6] C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction," in *ICDM*, 2007.

[7] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *LinkKDD*, 2004.

[8] J. R. Doppa, J. Yu, P. Tadepalli, and L. Getoor, "Chance constrained programs for link prediction," in *Analyzing Networks and Learning with Graphs (NIPS 2009)*, 2009.

[9] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Shölkopf, "Correcting sample selection bias by unlabeled data," in *NIPS*, 2006.

[10] J. Doppa, J. Yu, P. Tadepalli, and L. Getoor, "Link mining: A survey," in *SIGKDD Explorations*, 2005.

[11] L. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, 2001.

[12] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review Letters*, 2001.

[13] M. Bilgic, G. Namata, and L. Getoor, "Combining collective classification and link prediction," in *Workshop on Mining Graphs and Complex Structures (at ICDM)*, 2007.

[14] L. Getoor, N. Friedman, D. Koller, and B. Taskar, "Learning probabilistic models of relational structure," in *ICML*, 2001.

[15] ——, "Learning probabilistic models of link structure," *Journal of Machine Learning Research*, no. 3, pp. 679–707, 2002.

[16] O. Hassanzadeh and et al, "A framework for semantic link discovery over relational data," in *CIKM*, 2009.

[17] B. Taskar, P. Abbeel, and D. Kollerk, "Discriminative probabilistic models for relational data," in *UAI*, 2002.

[18] R. Raina, A. Ng, and D. Koller, "Constructing informative priors using transfer learning," in *Proceedings of Internatinal Conference on Machine Learning*, 2006.

[19] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[20] J. Tang, T. Lou, and J. M. Kleinberg, "Inferring social ties across heterogenous networks," in *Proceedings of WSDM*, 2012, pp. 743–752.

[21] G. Schwartz, "Esitmating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1979.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[23] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, vol. 31, no. 3, pp. 167–175, May 2003.

[24] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the $\ell_1$-ball for learning in high dimensions," in *Proc. of International Conference on Machine Learning*, Helsinki, Finland, 2008.