Transfer Learning to Infer Social Ties across Heterogeneous Networks

JIE TANG and TIANCHENG LOU, Tsinghua University
JON KLEINBERG, Cornell University
SEN WU, Tsinghua University

Interpersonal ties are responsible for the structure of social networks and the transmission of information through these networks. Different types of social ties have essentially different influences on people. Awareness of the types of social ties can benefit many applications, such as recommendation and community detection. For example, our close friends tend to move in the same circles that we do, while our classmates may be distributed into different communities. Though a bulk of research has focused on inferring particular types of relationships in a specific social network, few publications systematically study the generalization of the problem of predicting social ties across multiple heterogeneous networks.

In this work, we develop a framework referred to as TranFG for classifying the type of social relationships by learning across heterogeneous networks. The framework incorporates social theories into a factor graph model, which effectively improves the accuracy of predicting the types of social relationships in a target network by borrowing knowledge from a different source network. We also present several active learning strategies to further enhance the inferring performance. To scale up the model to handle really large networks, we design a distributed learning algorithm for the proposed model.

We evaluate the proposed framework (TranFG) on six different networks and compare with several existing methods. TranFG clearly outperforms the existing methods on multiple metrics. For example, by leveraging information from a coauthor network with labeled advisor-advisee relationships, TranFG is able to obtain an F1-score of 90% (8%–28% improvements over alternative methods) for predicting manager-subordinate relationships in an enterprise email network. The proposed model is efficient. It takes only a few minutes to train the proposed transfer model on large networks containing tens of thousands of nodes.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing**; • **Information systems** → *Information systems applications; World Wide Web*; • **Networks** → Network Types

Additional Key Words and Phrases: Social ties, social network, predictive model, social influence

ACM Reference Format:
Jie Tang, Tiancheng Lou, Jon Kleinberg, and Sen Wu. 2016. Transfer learning to infer social ties across heterogeneous networks. ACM Trans. Inf. Syst. 34, 2, Article 7 (April 2016), 43 pages.
DOI: http://dx.doi.org/10.1145/2746230

J. Tang is supported by the National High-Tech R&D Program (No. 2014AA015103), National Basic Research Program of China (No. 2014CB340506, No. 2012CB316006), Natural Science Foundation of China (No. 61222212), National Social Science Foundation of China (No. 13&ZD190), and a Huawei Research Grant. J. Kleinberg has been supported in part by a Google Research Grant, a Yahoo Research Alliance Grant, and NSF grants IIS-0910664, CCF-0910940, and IIS-1016099.
Authors' addresses: J. Tang, T. Lou, and S. Wu, Rm 1-308, FIT Building, Tsinghua University, Beijing, 100084. China; emails: jietang@tsinghua.edu.cn, tiancheng.lou@gmail.com, ronaldosen@gmail.com; J. Kleinberg, Computing and Information Science, Gates Hall, Cornell University, Ithaca, NY 14853, USA; email: kleinber@cs.cornell.edu.

DOI: http://dx.doi.org/10.1145/2746230

1. INTRODUCTION

In social networks, interpersonal ties generally fall into three categories: strong, weak, or absent. It is argued that more novel information flows to individuals through weak ties rather than strong ties [Granovetter 1973], while strong ties gather close friends into the same social circles [Krackhardt 1992]. Dunbar's number [Gladwell 2001] suggests that the number of people who can maintain stable social relationships lies between 100 and 230, commonly viewed as a value of 150. But the types of relationships would be very different. For example, among the 150, you may have five intimate friends, 15 family members, 35 colleagues (or classmates), and other acquaintances [Goncalves et al. 2011].

The rapid development recently of online social networks (e.g., Facebook, Twitter, LinkedIn, YouTube, and Slashdot) offers the opportunity to study the underlying patterns of social ties. For example, Facebook announced that it had hit a billion active accounts in October 2012. Tencent, one of the largest social networking services in China, has nearly 800 million users. People are connected via different types of social ties, and the influence between people varies largely with the type of social tie. For instance, in a mobile communication network, interpersonal ties can be roughly classified into four types: family, colleague, friend, and acquaintance. Colleagues have a strong influence on one's work, while friends have a strong influence on one's daily life. In an enterprise email network, where people are connected by sending/receiving emails to/from others, the ties between people can be categorized as manager-subordinate, colleague, and so forth. There is little doubt that behaviors in the email network are governed by the different types of relationships between senders and receivers.

Awareness of these different types of social relationships can benefit many applications. For example, if we can extract friendships between users from a mobile communication network, we can leverage the friendships for a "word-of-mouth" promotion of a new product. However, such information (relationship type) is usually unavailable in online networks. Users may easily add links (relationships) to others by clicking "friend request," "follow," or "agree" but do not often take the time to create labels for each relationship. Indeed, one survey of mobile phone users in Europe shows that only 16% of users had created contact groups on their mobile phones [Roth et al. 2010]; our preliminary statistics on LinkedIn data also shows that more than 70% of the connections have not been well labeled. In addition, the availability of the types of relationships in different networks is very unbalanced. In some networks, such as Slashdot, it might be easy to collect the labeled relationships (e.g., trust/distrust relationships between users). Facebook and Google+ provide a function to allow users to create "circles" (or "lists") [McAuley and Leskovec 2014]. However, in many other networks, it would be difficult to obtain the labeled information. Can we automatically predict the types of relationships in a social network? The difficulties of fulfilling the task vary largely in different networks. Can we leverage the available labeled relationships from one (source) network to help predict the types of relationships in another different (target) network? The problem is referred to as transfer link prediction across heterogeneous networks. Compared to traditional research on inferring social ties in one network (e.g., Wang et al. [2010], Crandall et al. [2010], and Tang et al. [2011]), this problem exhibits very different challenges:

First, **no common features:** as the two networks (source and target) might be very different, without any overlap, it is challenging to directly apply an existing transfer learning approach to this task. Figure 1 gives an example of link prediction across a product reviewer network derived from Epinions.com and a mobile communication network derived from a university. In the product reviewer network (called source network), we have labeled (trust and distrust) relationships and our goal is to leverage this

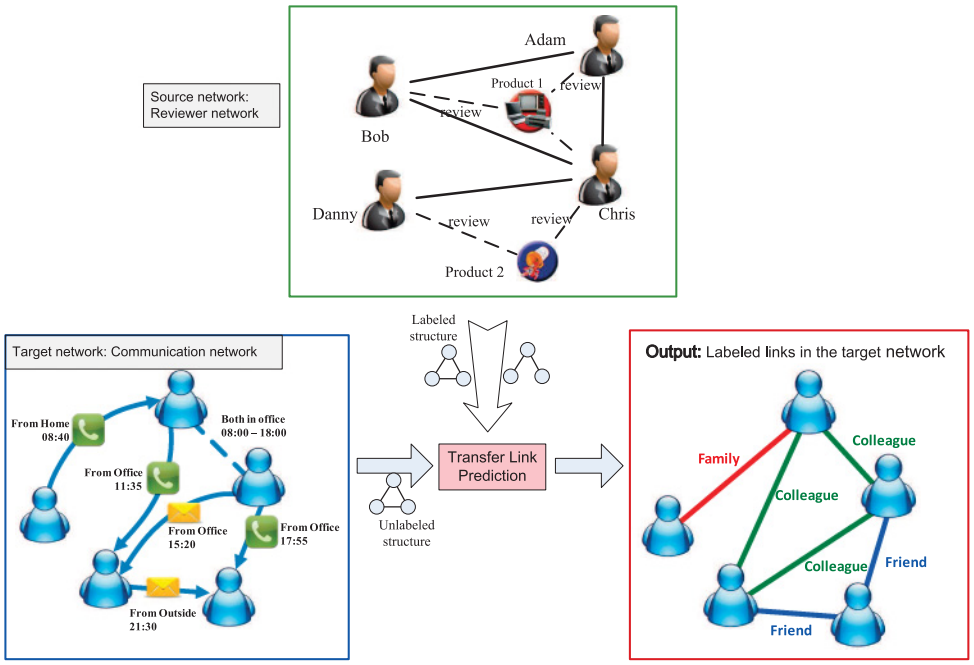


Fig. 1. Example of transfer link prediction across two heterogeneous networks: a reviewer network and a mobile communication network. In our problem setting, given a source network with sufficient labeled relationships and a target network with only a few relationships with labels, our goal is to leverage the labeled relationships/structures in the source network to help predict the types of relationships in the target network.

information to help predict the social relationship (family, friend, colleague) between users in the mobile network (called target network). The two networks are completely different. Fortunately, from the structural perspective, both networks share some general properties, such as obedience of power-law degree distribution [Barabási and Albert 1999] and satisfaction of social balance theory [Easley and Kleinberg 2010]. The fundamental challenge is how to transfer the structural information from the labeled source network to help prediction of the unlabeled structure in the target network.

Second, **network unbalance:** the scale of the input networks may be very unbalanced. For example, an online social network such as Facebook may consist of millions of users, while an enterprise email network may have only hundreds of users. Regarding both efficiency and effectiveness, it is impractical to simply train a model on one network and then directly apply it to the other network.

Third, **model generalization:** most existing models for predicting social ties are designed for specific networks. For example, Wang et al. [2010] only consider the coauthor network. How can one design a generalized framework to formalize the problem in a unified way?

In this work, we aim to conduct a systematic investigation for the problem of transfer link prediction across heterogeneous networks. To sum up, the contributions of this article are as follows:

—We precisely define the problem and propose a transfer-based factor graph (TranFG) model. The model incorporates social theories into a semisupervised learning framework, which can be used to transfer supervised information from a source network to help predict social ties in a target network.

—We present several active learning strategies to enhance the learning performance of the proposed model. To scale up the model to large social networks, we develop a distributed learning algorithm.
—We evaluate the proposed model on six different networks: Epinions, Slashdot, MobileU, MobileD, Coauthor, and Enron. We show that the proposed model can significantly improve the performance (on average +14% in terms of F1-measure) for predicting social ties across different networks comparing with several alternative methods.
—Our study also reveals several interesting phenomena for social science: (1) social balance is satisfied on friendship (or trust) networks, but not (<20% with a large variance) on user communication networks (e.g., mobile communication network); (2) users are more likely (up to +152% higher than chance) to have the same type of relationship with a user who spans a structural hole;[1] and (3) two strong ties are more likely to share the same type (15 times higher on Enron and Coauthor) than two weak ties.

This article is an extension of prior work [Tang et al. 2012]. Compared to the prior work, we have the following new contributions: (1) proposal of a new problem of *active transfer link prediction* and development of several effective strategies to address this problem; (2) development of a distributed learning algorithm for the proposed model framework; (3) investigation of a new social theory—Strong/Weak hypothesis—in various social networks; and (4) empirical evaluation of effectiveness of the newly proposed algorithm for active transfer link prediction and scalability performance of the distributed learning algorithm. Figure 2 shows a performance comparison of four algorithms for active link prediction on four different datasets. Clearly, the proposed Maximum Model Influence (MMI) performs much better than the other comparative algorithms. Figure 3 shows the scalability performance of the distributed learning algorithm for the TranFG model. The distributed learning algorithm is very efficient, achieving ∼9× speedup with 12 cores.

The rest of the article is organized as follows. Section 2 introduces the datasets used in this study. Section 3 formulates the problem. Section 5 presents our observations over the different networks. Section 6 explains the proposed model and describes the algorithm for learning the model. Section 7 presents the active learning algorithm to enhance the proposed model. Section 8 presents the distributed learning algorithms for the proposed model. Section 9 gives the experimental setup and results. Finally, Section 10 discusses related work, and Section 11 concludes.

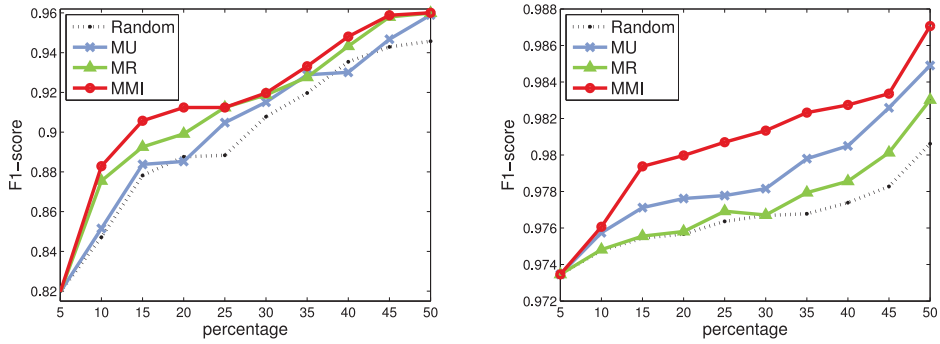
2. DATA DESCRIPTION

We study the problem of transfer link prediction on six different networks: Epinions, Slashdot, MobileU, MobileD, Coauthor, and Enron.

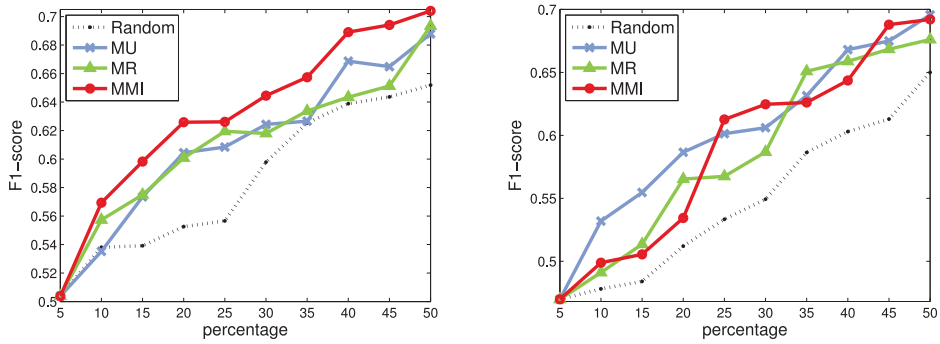
Epinions is a network of product reviewers. The dataset is from Leskovec et al. [2010b]. Each user on the site can post a review for any product and other users rate the review with trust or distrust. In this data, we created a network of reviewers connected with trust and distrust relationships. The dataset consists of 131,828 users and 841,372 relationships, of which about 85.0% are trust relationships; 80,668 users received at least one trust or distrust relationship. Our goal on this dataset is to predict the trust relationships between users.

Slashdot is a network of friends. Slashdot is a site for sharing technology-related news. In 2002, Slashdot introduced the Slashdot Zoo, which allows users to tag each other as “friends” (like) or “foes” (dislike). The dataset is composed of 77,357 users

[1]Structural hole is a concept from sociology [Burt 1992] and will be elaborated in the following sections.



(a) Epinions-to-Slashdot (b) Slashdot-to-Epinions



(c) Coauthor-to-MobileD (d) Enron-to-MobileD

Fig. 2. Performance of active learning. MMI is our proposed method; Random, MU, and MR are three comparison methods.

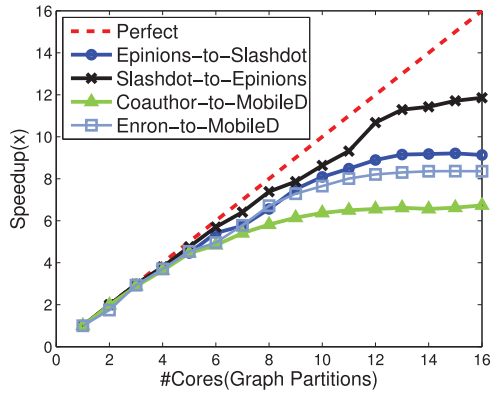


Fig. 3. Speedup versus #cores.

and 516,575 relationships, of which 76.7% are "friend" relationships. Our goal on this dataset is to predict the "friend" relationships between users.

MobileU is a network of mobile users. The dataset is from Eagle et al. [2009]. It consists of the logs of calls, Bluetooth scanning data, and cell tower IDs of 107 users during about 10 months. If two users communicated (by making a call and sending a

Table I. Statistics of Six Datasets

Relationship	Dataset	#Nodes	#Relationships (Positive)
Trust	Epinions	131,828	841,372 (715,166)
Friendship	Slashdot	77,357	516,575 (396,213)
Friendship	MobileU	107	5,436 (157)
Manager-subordinate	MobileD	232	3,567 (242)
Advisor-advisee	Coauthor	1,310	6,096 (514)
Manager-subordinate	Enron	151	3,572 (133)

Numbers in round brackets indicate "positive" relationships, respectively corresponding to trust, friend, friend, manager-subordinate, advisor-advisee, and manager-subordinate relationships in the six datasets.

text message) with each other or co-occurred in the same place, we create a relationship between them. In total, the data contains 5,436 relationships. Our goal is to predict whether two users have a friend relationship. For evaluation, all users are required to complete an online survey, in which 157 pairs of users are labeled as friends.

MobileD is a relatively larger mobile network of enterprise, where nodes are employees in a company and relationships are formed by calls and short messages sent between each other during a few months. In this mobile network, each user is labeled with his or her position (such as manager or ordinary employee) in the company. In total, there are 232 users (50 managers and 182 ordinary employees) and 3,567 relationships (including calling and texting messages) between the users. The objective here is to predict manager-subordinate relationships between users based on their mobile usage patterns.

Coauthor is a network of authors. The dataset, crawled from ArnetMiner.org [Tang et al. 2008], is composed of 815,946 authors and 2,792,833 coauthor relationships. In this dataset, we attempt to predict advisor-advisee relationships between coauthors. For evaluation, we created a smaller ground truth data using the following method: (1) collecting the advisor-advisee information from the Mathematics Genealogy project[2] and the AI Genealogy project[3] and (2) manually crawling the advisor-advisee information from researchers' homepages. Finally, we have created a dataset with 1,310 authors and 6,096 coauthor relationships, of which 514 are advisor-advisee relationships.

Enron is an email communication network [Diehl et al. 2007]. It consists of 136,329 emails sent among 151 Enron employees. Two types of relationships, that is, manager-subordinate and colleague, were annotated between these employees. Our goal on this dataset is to predict manager-subordinate relationships between users. There are in total 3,572 relationships, of which 133 are manager-subordinate relationships.

Table I lists statistics of the six networks. All datasets and codes used in this work are publicly available.[4] Please note that there are two slightly different prediction tasks: for the first three datasets (i.e., Epinions, Slashdot, and MobileU), our goal is to predict undirected relationships (friendships or trustful relationships), while for the other three datasets (i.e., Coauthor, Enron, and MobileD), our goal is to predict directed relationships (the source end has a higher social status than the target end, e.g., advisor-advisee relationships and manager-subordinate relationships). In principle, for each prediction task, any labeled network can be considered as a source network and any other network could be the target network. More specifically, for predicting undirected relationships, we tried all possible situations: Epinions (S) to Slashdot (T),

[2]http://www.genealogy.math.ndsu.nodak.edu.
[3]http://aigp.eecs.umich.edu.
[4]http://arnetminer.org/socialtieacross/.

Slashdot (S) to Epinions (T), Epinions (S) to MobileU (T), Slashdot (S) to MobileU (T), MobileU (S) to Slashdot (T), and MobileU (S) to Epinions (T). However, as the size of MobileU is much smaller than the other two networks, the performance was considerably worse. In the experiment, thus, we only report results of the first four pairs of networks. (Cf. Table III for details.) For predicting undirected relationships, we tried all possible transfer link prediction tasks and report results in Table IV.

3. PROBLEM DEFINITION

In this section, we first give several necessary definitions and then present the problem formulation. To simplify the explanation, we frame the problem with two social networks, a source network and a target network, although the generalization of this framework to a multiple-network setting is straightforward.

A social network can be represented as $G = (V, E)$, where V denotes a set of users and $E \subset V \times V$ denotes a set of relationships between users. In our problem, each relationship has a label to indicate the type of relationship. We may have the label information for some relationships, which is encoded as E^L, and for the other relationships encoded as E^U, we do not have the label information, where $E = E^L \cup E^U$.

Our general objective is to predict the types of relationships in E^U based on the available information in the social network. More specifically, let $\mathbf{X}$ be an $|E| \times d$ attribute matrix associated with relationships in E, with each row corresponding to a relationship, each column corresponding to an attribute, and the element x_{ij} denoting the value of the j^{th} attribute of relationship e_i. The label of relationship e_i is denoted as $y_i \in \mathcal{Y}$, where $\mathcal{Y}$ is the possible space of the labels (e.g., family, colleague, classmate). In principle, the label can be an arbitrary discrete value, but in this work, for easy explanation, we will focus on the binary case, for example, friend versus nonfriend in the Mobile network, advisor-advisee versus colleague in the coauthor network, or trust versus distrust in the Epinions network. Given this, we could have the following definition of a partially labeled network.

Definition 3.1. **Partially Labeled Network:** A partially labeled network is described as a five-tuple $G = (V, E^L, E^U, \mathbf{X}, Y)$, where V is a set of users, E^L is a set of labeled relationships, E^U is a set of unlabeled relationships, $\mathbf{X}$ is an attribute matrix associated with all relationships, and Y is a set of labels corresponding to the relationships in $E = E^L \cup E^U$, with $y_i \in Y$ denoting the type of relationship e_i.

When studying the link prediction problem in a single network, the input is a partially labeled network $G = (V, E^L, E^U, \mathbf{X}, Y)$, and the goal is to predict the unknown labels $\{y\}$ in Y. In this work, we study the link prediction problem across multiple networks. When considering two networks, the input to our problem consists of two partially labeled networks G_S (source network) and G_T (target network) with $|E_S^L| \gg |E_T^L|$ (with an extreme case of $|E_T^L| = 0$). Please note that the two networks may be totally different (with different sets of vertexes, i.e., $V_S \cap V_T = \varnothing$, and different attributes defined on relationships), such as a product reviewer network and a mobile communication network.

In different social networks, the relationship could be undirected (e.g., friendships in a mobile network) or directed (e.g., manager-subordinate relationships in an enterprise email network). To keep things consistent, if no ambiguity exists, we will concentrate on the undirected network, though we will also talk about directed networks. In the undirected network, if we predict a directed relationship label (e.g., the manager-subordinate relationship), then we consider each undirected relationship as two directed relationships. In addition, the label of a relationship may be static (e.g., the family-member relationship) or change over time (e.g., the manager-subordinate

relationship). In this work, we focus on static relationships. Thus, formally, we can define the following problem:

PROBLEM 1. **Transfer Link Prediction Across Social Networks:** *Given a source network G_S with abundantly labeled relationships and a target network G_T with a limited number of labeled relationships, the goal is to transfer learn a predictive function*

$$f : (G_T|G_S) \rightarrow Y_T$$

for predicting the types of relationships in the target network by leveraging the supervised information (labeled relationships) from the source network.

Without loss of generality, we assume that for each possible type y_i of relationship e_i, the predictive function will output a probability $p(y_i|e_i)$; thus, our task can be viewed as to obtain a triple $(e_i, y_i, p(y_i|e_i))$ for each relationship e_i in the social network.

It is worth noting that though we say there is only a limited number of labeled relationships in the target network, the labeled information is still very important. Without them, it is not clear what the learning task is in the target network, as the source and the target networks may have different prediction tasks. On the other hand, our assumption is that obtaining labeled relationships in the target network is often expensive. Hence, one more challenge is how to minimize the number of labeled relationships in the target network without hurting the prediction performance.

There are several key issues that make our problem formulation different from existing works on social relationship mining [Crandall et al. 2010; Diehl et al. 2007; Tang et al. 2011; Wang et al. 2010]. First, the source network and the target network may be very different, for example, a coauthor network and an email network. What are the fundamentally common factors that form the structure of the networks? Second, the labels of relationships in the target network and those of the source network could be different. How reliably can we predict the labels of relationships in the target network by using the information available in the source network? Third, as both the source and the target networks are partially labeled, the learning framework should consider not only the labeled information but also the unlabeled information.

4. BASIC PREDICTIVE MODELS

We first describe several basic predictive models for learning to predict social ties in social networks.

4.1. Link Prediction in Single Network

When considering a single network, the problem can be cast as a classification problem. For the input network $G = (V, E^L, E^U, \mathbf{X}, Y)$, each relationship e_i is associated with an attribute vector $\mathbf{x}_i$ and a label y_i indicating the type of relationship. Then the task is to find a classification model to predict the label of relationships in E^U. A straightforward idea is to use existing algorithms such as Support Vector Machines (SVMs) or Logistic Regression to train the classification model [Leskovec et al. 2010a]. If one further wants to consider the correlation among the predictive results $\{y\}$, then a graphical model such as Conditional Random Fields (CRFs) or Factor Graph Model (FGM) is preferable [Tang et al. 2011].

We use SVMs [Cortes and Vapnik 1995] as the example to explain how to predict social ties in a single social network. Given the labeled relationships in the input network G, we can construct a training dataset $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i$ is the attribute vector associated with relationship e_i and y_i corresponds to its label. There are generally two stages in the classification model, that is, learning and prediction. In learning, one attempts to find an optimal separating hyperplane that maximally

separates different categories of training examples. The hyperplane corresponds to an SVM classifier. It is theoretically guaranteed that the linear classifier obtained in this way has small generalization errors. Linear SVM can be further extended into nonlinear SVMs by using kernel functions such as Gaussian and polynomial kernels. In prediction, one can use the trained classification model to predict the unknown label of relationships in E^U. The process of applying logistic regression in the task is similar to that of Support Vector Machines.

The SVM-based method cannot model the correlation between the predictive results $\{y\}$ by assuming that they are independent of each other. In real social networks, this may be not the case. For example, in a coauthor network, predicting one coauthor relationship as an advisor-advisee relationship would correlate with the prediction result of another coauthor relationship. We will explain how we consider such correlation in Section 6.

4.2. Transfer Learning Across Networks

To transfer the knowledge from the source network into the target network, one could consider a transfer learning model. We briefly introduce a baseline transfer learning model, coclustering-based transfer learning (CoCC) [Dai et al. 2007a].

The basic idea of CoCC is to transfer the labeled information from a set D_i of "in-domain" documents to another set D_o of "out-of-domain" documents. CoCC uses coclustering as a bridge to propagate the labeled information from the in-domain to out-of-domain. Coclustering on out-of-domain data aims to simultaneously cluster the out-of-domain documents D_o and words $\mathcal{W}$ into $|\mathcal{C}|$ document clusters and k word clusters, respectively. Here $\mathcal{C}$ is the label space for the two domains.

Mathematically, CoCC tries to optimize the following loss function for coclustering-based learning:

$$I(D_o; \mathcal{W}) - I(\hat{D}_o; \hat{\mathcal{W}}) + \lambda \cdot (I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}})), \tag{1}$$

where $I(D_o; \mathcal{W})$ measures the mutual information between documents and words; $\hat{D}_o$ denotes the clustering of documents; $\hat{\mathcal{W}}$ denotes the clustering of words; $I(D_o; \mathcal{W}) - I(\hat{D}_o; \hat{\mathcal{W}})$ defines the loss in mutual information between documents and words before and after clustering; and, analogously, $I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}})$ defines the loss in mutual information between class labels $\mathcal{C}$ and words $\mathcal{W}$ before and after clustering. By minimizing this objective function and building a mapping between $\hat{D}_o$ and $\mathcal{C}$, CoCC is able to assign classes to documents in D_o according to the cluster membership, which enables the coclustering-based transfer learning.

Limitations. To deal with our problem, CoCC has three disadvantages. First, it makes an assumption that the labels of the in-domain and the out-of-domain data are drawn from the same label set. Second, it assumes that features in the two domains have a large overlap. Last, it is not easy to incorporate various correlation features such as the social-theory-based features into the CoCC model.

5. SOCIAL PATTERNS

We now engage in some high-level investigations of how different factors influence the formation of different social ties in different networks. Generally, if we consider predicting particular social ties in a specific network (e.g., mining advisor-advisee relationships from the Coauthor network [Wang et al. 2010]), we can define domain-specific features and learn a predictive model based on the labeled training data. The problem becomes very different when dealing with multiple heterogeneous networks, as the defined features in different networks may be significantly different. To this end,

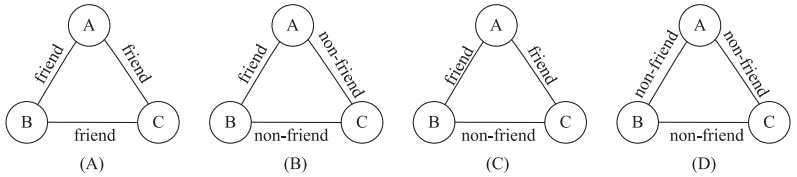


Fig. 4. Illustration of structural balance theory. (A) and (B) are balanced, while (C) and (D) are not balanced.

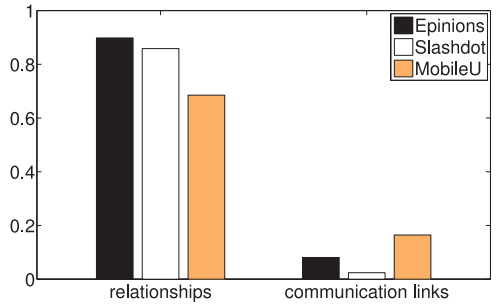


Fig. 5. Social balance. Probabilities of balanced triads in different networks based on communication links and friendships (or trustful relationships). Based on the communication links, different networks have very different balance probabilities (e.g., the balance probability in the mobile network is nearly 7 times higher than that of the Slashdot network). Based on friendships, however, the three networks have relatively similar probabilities.

we connect our problem to several basic social theories and focus our analysis on the network-based correlations via the following statistics:

(1) *Social balance [Easley and Kleinberg 2010].* How is the social balance property satisfied and correlated in different networks?
(2) *Structural hole [Burt 1992; Lou and Tang 2013].* Would structural holes have a similar pattern in different networks?
(3) *Social status [Davis and Leinhardt 1972; Guha et al. 2004; Leskovec et al. 2010b].* How do different networks satisfy the properties of social status?
(4) *"Two-step flow" [Lazarsfeld et al. 1944].* How do different networks follow the "two-step flow" of information diffusion?
(5) *Strong/weak tie hypothesis [Granovetter 1973; Krackhardt 1992].* How are the strong tie and weak tie hypotheses correlated in different networks?

Social Balance. Social balance theory suggests that people in a social network tend to form into a balanced network structure. Figure 4 shows such an example to illustrate the structural balance theory over triads, which is the simplest group structure to which balance theory applies. For a triad, the balance theory implies that either all three of these users are friends or only one pair of them are friends. Figure 5 shows the probabilities of balanced triads of the three undirected networks (Epinions, Slashdot, and MobileU). The probability of balanced triads based on underlying (communication) links is calculated by

$$P(v_i, v_j, v_k \text{ have 1 or 3 underlying/communication links} | e_{ij}, e_{jk}, e_{ki} \in E), \quad (2)$$

and the probability based on friendships (or trust relationships) is calculated by

$$P(v_i, v_j, v_k \text{ have 1 or 3 friend/trust relationships} | e_{ij}, e_{jk}, e_{ki} \in E). \quad (3)$$

In each network, we compare the probability of balanced triads based on communication links and that based on friendships (or trust relationships). For example,

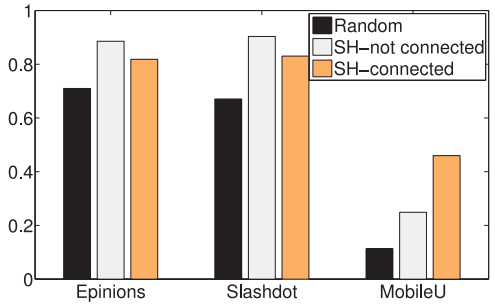


Fig. 6. Structural hole. SH-connected and SH-not connected respectively represent the probability that two connected and two disconnected users (A and B) have the same type of relationship with user C, conditioned on user C spanning a structural hole. "Random" indicates the average probability that two users (A and B) have the same type of relationship with user C regardless of whether C spans a structural hole. It is clear that users are more likely (on average +70% higher than chance) to have the same type of relationship with C if C spans a structural hole.

in the mobile network, the communication links include making a call or sending a message between users. We find it interesting that different networks have very different balance probabilities based on the communication links; for example, the balance probability in the mobile network is nearly 7 times higher than that of the Slashdot network, while based on friendships (or trustful relationships) the three networks have relatively similar balance probabilities (with a maximum of +28% difference).

Structural Hole. Roughly speaking, a user is said to span a *structural hole* in a social network if he or she is linked to people in parts of the network that are otherwise not well connected to one another [Burt 1992]. Such a user is also called a *structural hole spanner* [Lou and Tang 2013]. Arguments based on structural holes suggest that there is an informational advantage to have friends in a network who do not know each other. A sales manager with a diverse range of connections can be considered as a structural hole spanner, with a number of potentially *weak ties* [Granovetter 1973] to individuals in different communities. More generally, we can think about websites such as eBay as spanning structural holes, in that they facilitate economic interactions between people who would otherwise not be able to find each other.

Our idea here is to test if a structural hole spanner tends to have the same type of relationship with the other users. We first employ a simple algorithm to identify structural hole spanners in a network. Following the informal description of structural holes [Burt 1992], for each node, we count the number of pairs of neighbors who are not directly connected. All users are ranked based on the number of pairs, and the top 1% of structural hole spanners[5] with the highest numbers are viewed as structural hole spanners in the network. Figure 6 shows the probabilities that two users (A and B) have the same type of relationship with another user (say, C), conditioned on whether user C spans a structural hole. "Random" indicates the average probability that two users (A and B) have the same type of relationship with user C regardless of whether C spans a structural hole.

We have two interesting observations: (1) users are more likely to have the same type of relationship with C if C spans a structural hole (e.g., in Epinions, the probability that two users have the same type of relationship with a structural hole spanner is +20% higher than the random case), and (2) disconnected users are more likely than

[5]This is based on the observation that less than 1% of the Twitter users produce 50% of its content [Wu et al. 2011].

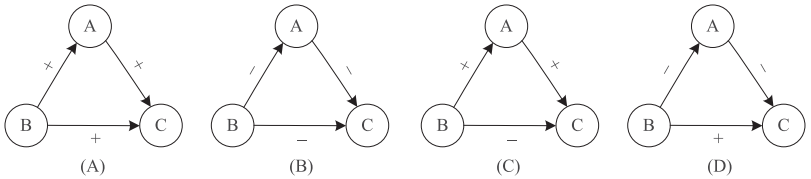
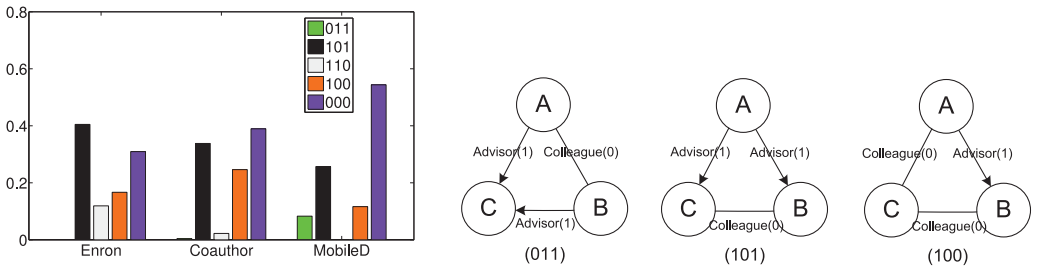


Fig. 7. Illustration of status theory. (A) and (B) satisfy the status theory, while (C) and (D) do not satisfy the status theory. Here, positive "+" denotes that the target node has a higher status than the source node, and negative "−" denotes that the target node has a lower status than the source node. In total there are 16 different cases.



(a) Social status (b) Example triadic codes

Fig. 8. Distribution of the five most frequent formations of triads with social status. Given a triad (A, B, C), let us use 1 to denote the advisor-advisee relationship and 0 to denote the colleague relationship. The number 011 denotes that A and B are colleagues, B is C's advisor, and A is C's advisor.

connected users to have the same type of relationship with a user classified as spanning a structural hole. One exception is the mobile network, where most mobile users in the dataset are university students and thus friends frequently communicate with each other.

Social Status. Another social psychological theory is the theory of status [Davis and Leinhardt 1972; Guha et al. 2004; Leskovec et al. 2010b]. This theory is based on the directed relationship network. Suppose each directed relationship is labeled by a positive sign "+" or a negative sign "−" (where sign "+"/"−" denotes the target node has a higher/lower status than the source node). Then status theory posits that if, in a triangle on three nodes (called triad), we take each negative relationship, reverse its direction, and flip its sign to positive, then the resultant triangle (with all positive relationship signs) should be acyclic. Figure 7 illustrates four examples. The first two triangles satisfy the status ordering and the latter two do not satisfy it. We conducted an analysis on the Coauthor and the Enron networks, where we aim to find directed relationships (advisor-advisee and manager-subordinate). We found that nearly 99% of triads in the two networks satisfy the social status theory, which was also validated in Leskovec et al. [2010b]. We investigate more by looking at the distribution of different forms of triads in the two networks. Specifically, there are in total 16 different forms of triads [Leskovec et al. 2010b]. We select the five most frequent forms of triads in the two networks. For easy understanding, given a triad (A, B, C), we use 1 to denote the advisor-advisee relationship and 0 to denote the colleague relationship, and three consecutive numbers 011 to denote A and B are colleagues, B is C's advisor, and A is C's advisor. It is interesting to see that although the three networks (Coauthor, Enron, and MobileD) are totally different, the three types of triads with the highest probabilities are the same, that is, 000, 100, and 101 ("101" indicates that the statuses of the three users in the triad are, respectively, high, low, and high). Figure 8 plots the five triads

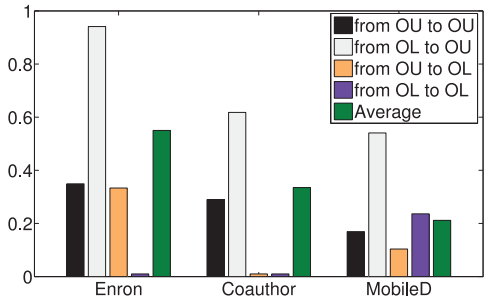


Fig. 9. Opinion leader. OL - Opinion leader; OU - Ordinary user. Probability that two types of users have a directed relationship (from higher social status to lower status, i.e., manager-subordinate relationship in Enron and advisor-advisee relationship in Coauthor). Average indicates the average probability that two random users have a relationship of high-status to lower-status user. It is clear that opinion leaders (detected by PageRank) are more likely to have a higher social status than ordinary users.

with the highest probabilities in the three networks. In practice, some patterns such as 111 seem to be unreasonable. However, there still exist some cases in real networks, for various reasons. In our problem, we are interested in understanding to what extent this case is unreasonable and how different networks correlate on this social pattern. The pattern correlation will be used to transfer information from the source network to the target network.

Opinion Leader. The two-step flow theory was first introduced in Lazarsfeld et al. [1944] and later elaborated in the literature [Katz 1957; Katz and Lazarsfeld 1955]. The theory suggests that ideas (innovations) usually flow first to *opinion leaders* and then from them to a wider population. In the enterprise email network, for example, managers may act as opinion leaders to help spread information to subordinates.

Our basic idea here is to examine whether "opinion leaders" are more likely to have a higher social status (manager or advisor) than ordinary users. To do this, we first categorize users into two groups (opinion leaders and ordinary users) by PageRank.[6] There is considerable research on opinion leader. For example, Song et al. [2007] present a PageRank-like algorithm (referred to as InfluenceRank) to identify opinion leaders, and Wang et al. [2011] propose an algorithm to find kernel members (elite users) in a social network. However, designing new measures for finding opinion leaders is beyond the focus of this article; hence, we adopt the simple intuitive measure PageRank to select opinion leaders. With PageRank, according to the network structure, we select the top 1% of users who have the highest PageRank scores as opinion leaders and the rest as ordinary users. Then, we examine the probabilities that two users (A and B) have a directed social relationship (from higher social-status user to lower social-status user), such as advisor-advisee relationship. Figure 9 shows some interesting discoveries. First, in all of the Enron, Coauthor, and MobileD networks, opinion leaders (detected by PageRank) are more likely (+71%–+156%) to have a higher social status than ordinary users. Second and also more interestingly, in Enron, it is likely that ordinary users have a higher social status than opinion leaders. Its average likelihood is much larger (30 times) than that in the Coauthor network. The reason might be that in the enterprise email network, some managers may be inactive, and most management-related communications were done by their assistants.

Strong Tie Versus Weak Tie. Interpersonal ties generally come in three varieties: strong, weak, or absent. The strong tie hypothesis implies that one's close friends tend

[6]PageRank is an algorithm to estimate the importance of each node in a network [Page et al. 1999].

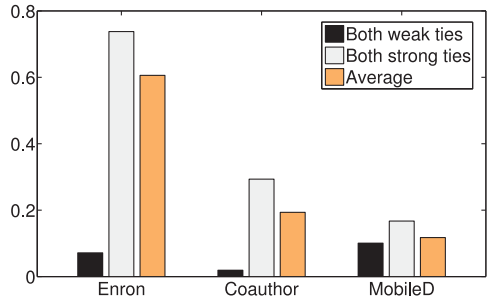


Fig. 10. Strong tie versus weak tie. Probabilities of two social ties sharing the same type, conditioned on whether the two social ties are strong or weak. Average indicates the probability that two random social ties share the same type. It is clear that on all three datasets, two strong ties result in a higher likelihood to share the same type than chance, while two weak ties are much more uncertain.

to move in the same circles that he or she does. Acquaintances, by contrast, constitute a more uncertain and dynamic social relationship. Thus, intuitively, a user may have similar types of relationships with friends of strong ties and more diverse relationships with friends of weak ties. Thus, we examine how the types of social ties are correlated with their strength.

For simplicity, we quantify the strength of a social tie in the following ways.[7] In the Coauthor network, for each relationship, we count the number of publications coauthored by the linked two authors. In the MobileD network, the strength of each social tie is quantified by the number of calls/text messages made between the linked two persons. In Enron, the strength is estimated by the number of emails sent between two users.[8] Then, we rank all social ties according to the strength and take the top one-third as strong ties and the rest as weak ties.

Figure 10 shows the likelihood of two social ties sharing the same type, conditioned on whether the two social ties are strong or weak. It clearly illustrates that in all the datasets, two strong ties result in a higher likelihood to share the same type than chance, while two weak ties are much more uncertain: the likelihood of two weak ties sharing the same type is merely one-ninth of that of two random social ties on both Enron and Coauthor.

Summary. According to these statistics, we have the following intuitions:

(1) Probabilities of balanced triads based on communication links are very different in different networks, while the balance probabilities based on friendships (or trustful relationships) are similar to each other.
(2) Users are more likely (+25%–152% higher than chance) to have the same type of relationship with a user who spans a structural hole.
(3) Most triads (99%) satisfy properties of the social status theory. For the five most frequent formations of triads, the Coauthor, the Enron, and the MobileD networks share a similar distribution.
(4) Opinion leaders are more likely (+71%–156% higher than chance) to have a higher social status than ordinary users.

[7]For a more theoretical study of quantifying social strength (or social influence), please refer to Tang et al. [2009].
[8]We could also estimate the strength of social ties on Epinions and Slashdot, for example, by the number of interactions (messages/replies sent) between the two users. However, due to the lack of the interaction information in the two networks, we only test the strong tie and weak tie hypotheses on Enron, Coauthor, and MobileD.

(5) Two strong ties have a higher likelihood (+22%–52%) to share the same type than chance, while two weak ties are much more uncertain (one-ninth of the likelihood of two random social ties to share the same type).

Based on these observations, we accordingly define features in the transfer learning model introduced in Section 4. The importance of different features will be determined by the learning model. Roughly speaking, if a social-theory-based feature has a similar pattern (e.g., social balance has a similar distribution over two networks), then the feature would have a high weight in the learned model; otherwise, it will have a small weight.

6. MODEL FRAMEWORK FOR TRANSFER LINK PREDICTION

We propose a transfer-based framework (TranFG). The basic idea in the framework is to incorporate the social theories into a factor graph model for learning and predicting the types of social relationships across different networks.

6.1. Probabilistic Factor Graph Model

Let us begin with a brief introduction of the graphical model. The major difference of the graphical model from the classification-based model (such as SVM) lies in that the graphical model can model the correlation between the prediction results by incorporating "edge" features (also called correlation features). In general, there are two types of graphical models: directed graphical model and undirected graphical model [Wainwright and Jordan 2008]. In this work, we consider the undirected graphical model. In the undirected graphical model, the graph is formed by a collection of variables $Y = \{y_i\}_{i=1,\ldots,n}$ and a collection of correlations between these variables.[9] According to the graphical structure, the probability distribution over the graph can be factorized as a collection of functions defined on the *cliques* of the graph. A clique c is a fully connected subset of the variables Y_c in the graph. For example, if a clique c consists of two vertices in the graph, then Y_c indicates the set of the two corresponding variables (e.g., $Y_c = \{y_{c1}, y_{c2}\}$). According to the theory of the undirected graphical model [Hammersley and Clifford 1971], we could associate a function with each clique, that is, $f(Y_c)$. Given this, the probability distribution of the graph is factorized as

$$P(Y) = \frac{1}{Z}\prod_c f(Y_c), \quad Z = \sum_{Y_c}\prod_c f(Y_c), \tag{4}$$

where Z is a normalization factor, also viewed as a constant chosen to ensure that the distribution is normalized so that the sum of the probabilities equals 1.

Factor graph presents an alternative way to describe the (undirected) graphical model, with more emphasis on the factorization of the distribution [Kschischang et al. 2001]. We will use the factor graph for the explanation in the following sections. Basically, the process of applying a graphical model to predict social ties also consists of two stages: training and prediction. Usually in a graphical model, one attempts to maximize the conditional probability of labels associated with the relationships given attributes of the corresponding relationships in the training data, that is, $P(Y^L|\mathbf{X}^L, G)$. Thus, in training, it attempts to find a parameter configuration that maximizes the conditional probability on labeled relationships E^L, and in prediction, it tries to find a setting of labels Y^U for the unlabeled relationships E^U to maximize the conditional probability $P(Y^U|\mathbf{X}^U, G)$. Directly maximizing the conditional probability $P(Y^L|\mathbf{X}^L, G)$ is often intractable when the graph structure contains cycles. Factor graph is a method

[9]In existing literatures (e.g., Wainwright and Jordan [2008]), the variables are also denoted as vertices in the graphical model and the correlations are denoted as edges between the vertices.

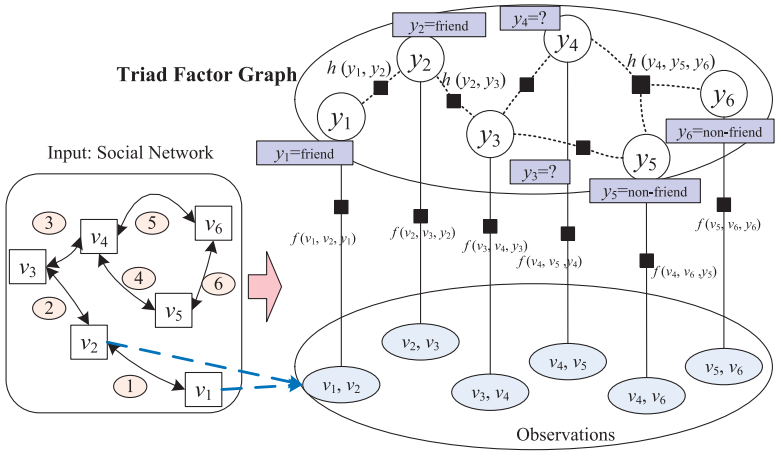


Fig. 11. Graphical representation of the triad factor graph (TriFG) model. Each ellipse indicates a relationship; for example, (v_1, v_2) represents a relationship between v_1 and v_2. Notation $\mathbf{x}_1$ indicates the attribute vector associated with the relationship (v_1, v_2). For each relationship, we use a corresponding latent variable y denoted as a circle to represent its type. Each black square denotes a factor function that takes the associated variables as input and outputs a real number.

to factorize the "global" probability as a product of "local" factor functions, each of which depends on a subset of the variables in the graph [Kschischang et al. 2001].

Equation (4) can be explained in terms of the factor functions. Each $f(Y_c)$ represents a factor defined over variables included in Y_c. Such a representation also has a property in terms of conditional independence among subsets of variables, which is also referred to as the Markov property in the undirected graphical model. Here we give a brief introduction of the concept of conditional independence. Interested readers may refer to Lauritzen [1996]. For undirected graphical models, if two variables are disconnected in the graphical model, then we say the two variables are independent. Suppose y_i, y_j, and y_k are an arbitrary triple of three variables. Let y_i be connected to y_j and y_j be connected to y_k. We say that variables y_i and y_k are conditional independent given y_j if variables y_i and y_k are only connected via y_j (i.e., disconnected without variable y_j). This property also can be generalized from multiple variables to multiple disjoint subsets of variables.

Regarding the graphical structure of the probabilistic factor graph model in our problem, we consider each relationship as an observation variable x in the graphical model. We associate each observation variable with a latent variable y and define a local factor function between each observation variable and its associated latent variable. We also define the correlation between the latent variables. Please note that the correlation can be defined among multiple relationships; for example, a triad function is defined to capture the correlation of a social balance structure. The correlations between (or among) relationships constitutes the graphical structure of the factor graph model. More specifically, if we define a correlation function between two latent variables, then an edge is created between the two variables, and if we define a correlation function of social balance, then a triadic function structure is constructed. Figure 11 shows the graphical representation of the triad factor graph (TriFG) model [Lou et al. 2013]. Each ellipse indicates a relationship; for example, (v_1, v_2) represents a relationship between v_1 and v_2. More accurately, in the model, it represents the attribute vector $\mathbf{x}$ associated with the relationship. Each circle is a variable y indicating the type of the corresponding relationship. Each black square denotes a factor function that

takes the associated variables as input and outputs a real number. The factor function $f(\mathbf{x}_1, y_1)$ is defined on attributes associated with relationship (v_1, v_2) (or e_1) and the factor $h(y_1, y_2)$ is defined to capture the correlation between y_1 and y_2. Specifically, if we only consider pairwise correlation, that is, correlation between pairwise relationships, then a pairwise factor graph model can be constructed accordingly [Kschischang et al. 2001; Tang et al. 2011]. Its underlying principle is also similar to the conditional random field [Lafferty et al. 2001], a conditional variation of the Markov random field. We consider both pairwise correlations and triads as cliques in our factor graph model, in that several social theories (such as social balance and social status) that we discussed in Section 5 are based on triads. In this case, the basic pairwise factor graph is extended as a triad factor graph (TriFG). As the example in Figure 11 shows, we could define six attribute factor functions, $f(\mathbf{x}_1, y_1)$, $f(\mathbf{x}_2, y_2)$, $f(\mathbf{x}_3, y_3)$, $f(\mathbf{x}_4, y_4)$, $f(\mathbf{x}_5, y_5)$, and $f(\mathbf{x}_6, y_6)$; four pairwise correlation factor functions, $h(y_1, y_2)$, $h(y_2, y_3)$, $h(y_3, y_4)$, $h(y_3, y_5)$; and one triadic correlation factor function, $h(y_4, y_5, y_6)$. According to the factorization principle in the factor graph [Kschischang et al. 2001], we could use the product of these factor functions to represent the joint probability $P(Y|\mathbf{X}, G)$ as follows:

$$P(Y|\mathbf{X}, G) = \frac{1}{Z} \prod_{i=1}^{|E|} f(\mathbf{x}_i, y_i) \prod_{c \in G} f(Y_c), \tag{5}$$

where $f(\mathbf{x}_i, y_i)$ represents a factor function defined according to the attributes $\mathbf{x}_i$, c is a clique on the graph (e.g., a triad (y_4, y_5, y_6)), and Y_c is a set of label variables included in the clique c. $f(Y_c)$ represents a factor function defined to capture the correlation between (among) all variables in Y_c. Finally, Z is a normalization factor, which is the summation of all possible values for Y. Formally, it can be written as

$$Z = \sum_{y} \prod_{i=1}^{|E|} f(\mathbf{x}_i, y_i) \prod_{c \in G} f(Y_c).$$

There are different ways to instantiate the two factor functions $f(\mathbf{x}_i, y_i)$ and $f(Y_c)$. A widely used method is to define them as an exponential-linear function, that is,

$$f(y_i, \mathbf{x}_i) = \frac{1}{Z_1} \exp\left\{ \sum_{j=1}^{d} \alpha_j g_j(x_{ij}, y_i) \right\} \tag{6}$$

$$Z_1 = \sum_{y_i} \exp\left\{ \sum_{j=1}^{d} \alpha_j g_j(x_{ij}, y_i) \right\} \tag{7}$$

$$f(Y_c) = \frac{1}{Z_2} \exp\left\{ \sum_{c} \sum_{k} \mu_k h_k(Y_c) \right\} \tag{8}$$

$$Z_2 = \sum_{Y_c} \exp\left\{ \sum_{c} \sum_{k} \mu_k h_k(Y_c) \right\}, \tag{9}$$

where Z_1 and Z_2 are two normalization factors to ensure that the sum of the distributions equals 1; Equation (6) indicates that we define a feature function $g_j(x_{ij}, y_i)$ for each attribute x_{ij} associated with relationship e_i and α_j is the weight of the j^{th} attribute. It can be defined as either a binary function or a real-valued function. For example,

for predicting advisor-advisee relationships from the publication network [Wang et al. 2010], we can define a real-valued feature function as the difference of years when authors v_i and v_j respectively published their first paper. Equation (8) represents that we define a set of correlation feature functions $\{h_k(Y_c)\}_k$ over each clique Y_c in the network. Here μ_k is the weight of the k^{th} correlation feature function. The simplest clique represents a pairwise correlation $h_k(y_i, y_j)$ between two relationships e_i and e_j.

By integrating Equations (6) and (8) into Equation (5), we can obtain the following log-likelihood objective function:

$$\mathcal{O}(\theta) = \log p(Y|\mathbf{X}, G) = \sum_{i}^{|E|} \sum_{j=1}^{d} \alpha_j g_j(x_{ij}, y_i) + \sum_{c} \sum_{k} \mu_k h_k(Y_c) - \log Z. \tag{10}$$

Here, we use θ to denote all unknown parameters, that is, $(\{\alpha\}, \{\mu\})$; Z_1 and Z_2 are combined into Z, which can be viewed as a constant to ensure the sum of the distributions equals 1.

If we are given a network G with labels Y, learning the predictive model is to estimate a parameter configuration $\theta^\star = (\{\alpha\}, \{\mu\})$ to maximize the log-likelihood objective function $\mathcal{O}(\theta) = \log P_\theta(Y|\mathbf{X}, G)$, that is,

$$\theta^\star = \arg\max \mathcal{O}(\theta). \tag{11}$$

Note and Limitation. It is worth noting that in practice, both training and prediction will be conducted on the same network, which is different from the traditional learning setting, where several graphs are fed for training and other graphs are used for prediction. In this case, it is necessary to design a method that can perform training and prediction simultaneously on the same input network. In the following subsection, we will consider the factor graph as a partially labeled network and perform the learning and prediction tasks together. Another limitation of the factor graph model is that it assumes all examples in the training and the prediction have the same feature distribution. This makes it difficult to directly apply the factor graph model to our problem of predicting social ties across social networks, as in our problem the source network and the target network could be very different without any common features in X.

6.2. Transfer-Based Factor Graph (TranFG) Model

Now we discuss how to design a factor graph model for learning to predict social ties across different networks. The basic idea is to leverage the power of the graphical model that can model the correlation among the prediction results and at the same time avoid the limitations in the existing transfer learning model. More specifically, we focus on learning a predictive model with two heterogeneous networks (a source network G_S and a target network G_T). Straightforwardly, we can define two separate objective functions for the two networks. The challenge is how to bridge the two networks so that we can transfer the labeled information from the source network to help predict social ties in the target network. As the source and the target networks may be from arbitrary domains, it is difficult to define correlations between them based on prior knowledge.

To this end, we propose a transfer-based factor graph (TranFG) model. Our idea is based on the fact that the social theories we discussed in Section 5 are general over all networks. Intuitively, we can leverage the correlation to the extent that different networks satisfy each of the social theories to transfer. Then we could define features based on the social theories and simultaneously optimize the objective function over the two networks. By incorporating the social theories into our predictive model, we define the following log-likelihood objective function over the source and the target

networks:

$$\begin{aligned}\mathcal{O} &= \mathcal{O}_S(\alpha, \mu) + \mathcal{O}_T(\beta, \mu) \\ &= \sum_{i=1}^{|E_S|}\sum_{j=1}^{d} \alpha_j g_j(x_{ij}^S, y_i^S) + \sum_{i=1}^{|E_T|}\sum_{j=1}^{d'} \beta_j g'_j(x_{ij}^T, y_i^T) \\ &\quad + \sum_k \mu_k \left(\sum_{c\in G_S} h_k(Y_c^S) + \sum_{c\in G_T} h_k(Y_c^T) \right) - \log Z,\end{aligned} \tag{12}$$

where d and d' are numbers of attributes in the source network and the target network, respectively. In this objective function, the first term and the second term define the likelihood, respectively, over the source network and the target network, while the third term defines the likelihood over common features defined in the two networks; function $h_k(Y_c)$ is a common feature defined according to the social theory and μ_k is the weight (importance) of the corresponding feature. According to the definition, if a common feature has a similar pattern (e.g., social balance has a similar distribution over two networks), then the feature would have a high weight μ; otherwise, it will have a small weight. Such a definition also implies that attributes of the two networks can be entirely different as they are optimized with different parameters $\{\alpha\}$ and $\{\beta\}$, while the information transferred from the source network to the target network is the importance of common features defined according to the social theories.

Another issue, as mentioned before, is that the input network might be partially labeled; thus, it is necessary to perform training and prediction simultaneously on the same input network. We will discuss this issue in Section 6.4.

6.3. Factor Function Definition

We now turn to the definition of the factor function. There are mainly two types of features. The first type of features are domain-specific features, which are different in different networks. For example, in the mobile network, we could define a feature for each relationship as the number of calls made between two users. The appendix gives more details on how we define these features for each network.

The second type of features are common features defined according to the social theories. This is one of the contributions in this work.

Social balance: Four (real-valued) features are defined to respectively represent the proportions of the four types of (un)balanced triangles in a network.

Social status: According to the social status theory [Leskovec et al. 2010b], there are 16 different triads in total. In our cases, we have found that only seven of them exist in our networks. We define seven (real-valued) features (011, 101, 110, 100, 000, 111, and −111) over triads to respectively represent the probabilities of the formations of triads.

Structural hole: We define relationship-correlation-based features, that is, the correlation between two relationships e_i and e_j. For example, if both B and C are connected with A, who is identified as a structural hole spanner, then two correlation features are defined for the two relationships ($A-B$ and $A-C$): one for B and C, who are connected, and the other for B and C, who are not connected.

Opinion leader: Four (binary) features are defined for each relationship ($A-B$). Two of them represent when both A and B are opinion leaders or ordinary users, while the other two features represent when one is an opinion leader and the other is an ordinary user.

ALGORITHM 1: Learning Algorithm for TranFG

Input: a source network G_S, a target network G_T, and the learning rate η
Output: estimated weights $\theta = (\{\alpha\}, \{\beta\}, \{\mu\})$ respectively for different types of features

Initialize $\theta \leftarrow 0$;
Perform statistics according to social theories;
Construct social-theory-based features $h_k(Y_c)$, here Y_c is a set of variables defined on clique c;
repeat
  Step 1: Perform LBP to calculate marginal distribution of unknown variables in the source network $P(y_i|x_i, G_S)$;
  Step 2: Perform LBP to calculate marginal distribution of unknown variables in the target network $P(y_i|x_i, G_T)$;
  Step 3: Perform LBP to calculate the marginal distribution of clique c, i.e., $P(y_c|\mathbf{X}_c^S, \mathbf{X}_c^T, G_S, G_T)$;
  Step 4: Calculate the gradient of μ_k according to Equation (13) (for α_j and β_j with a similar formula);
  Step 5: Update parameter μ_k (as well as α_j and β_j) with the learning rate η:

$$\mu_{\mathbf{k}}{}^{\text{new}} = \mu_{\mathbf{k}}{}^{\text{old}} + \eta \cdot \frac{\mathcal{O}(\theta)}{\mu_k}$$

until *Convergence*;

Strong/weak ties: Three (real-valued) features are defined to represent when two relationships are both strong ties, both weak ties, or one is a strong tie and the other is a weak tie.

Finally, we define four (real-valued) balance-based features, seven (real-valued) status-based features, four (binary) features for opinion leader, six (real-valued) correlation features for structural hole, and three (real-valued) features for strong/weak ties.

6.4. Model Learning and Prediction

The last issue is to learn the TranFG model and to predict the type of unknown relationships in the target network. Learning the TranFG model is to estimate a parameter configuration $\theta = (\{\alpha\}, \{\beta\}, \{\mu\})$ to maximize the log-likelihood objective function $\mathcal{O}(\alpha, \beta, \mu)$. The learning algorithm is summarized in Algorithm 1.

There are two challenges. First, as the network structure in the social network can be arbitrary (may contain cycles), it is intractable to obtain an exact solution of the objective function using existing methods such as Junction Tree [Wiegerinck 2000]. A number of approximate algorithms can be considered, such as Loopy Belief Propagation (LBP) [Murphy et al. 1999] and Mean field [Xing et al. 2003]. One can also derive the dual form of the original optimization problem (Equation (12)) and use the projected subgradient method [Komodakis et al. 2011] to solve the problem. We chose Loopy Belief Propagation due to its ease of implementation and effectiveness. Specifically, we approximate the marginal distribution $P_{\theta_k}(Y_c|.)$ using LBP. With the marginal probabilities, the gradient can be obtained by summing over all triads. Theoretically, the algorithm does not guarantee a convergence and may result in local maximum, but in practice its performance is good. We will empirically study the effectiveness and efficiency of the algorithm in the experimental section.

The other challenge is that both input networks may be partially labeled. To deal with this, we use the labeled relationships to infer the unknown labels. Here $Y|Y^L$ denotes a labeling configuration Y inferred from the known labels. Technically, we use an extra LBP process to infer the marginal probability of unlabeled relationships. In Algorithm 1, Step 1 is used to calculate marginal distributions of unknown

relationships in the source network, and Step 2 is used to calculate marginal distributions of unknown relationships in the target network, respectively. A similar learning algorithm was first introduced in Tang et al. [2011].

We use a gradient descent method (or a Newton-Raphson method) to solve the objective function. Specifically, we first write the gradient of each unknown parameter (α, β, μ) with regard to the objective function (we use α as the example to derive its gradient w.r.t. the objective function):

$$\begin{aligned}
\frac{\partial \mathcal{O}(\theta)}{\partial \alpha_j} &= \frac{\partial \left(\sum_{i=1}^{|E_S|} \sum_{j=1}^{d} \alpha_j g_j(x_{ij}^S, y_i^S) - \log Z \right)}{\partial \alpha} \\
&= \frac{\partial \left(\sum_{i=1}^{|E_S|} \sum_{j=1}^{d} \alpha_j g_j(x_{ij}^S, y_i^S) - \log \sum_y \sum_{i=1}^{|E_S|} \sum_{j=1}^{d} \alpha_j g_j(x_{ij}^S, y_i^S) \right)}{\partial \alpha} \\
&= \sum_{i=1}^{|E_S|} \alpha_j g_j(x_{ij}^S, y_i^S) - \frac{\sum_{y_i^S} \sum_{i=1}^{|E_S|} \alpha_j g_j(x_{ij}^S, y_i^S)}{\sum_{y_i^S} \sum_{i=1}^{|E_S|} \sum_{j=1}^{d} \alpha_j g_j(x_{ij}^S, y_i^S)} \\
&= \mathbb{E}[g_j(x_{ij}^S, y_i^S)] - \mathbb{E}_{P_{\alpha_j}(y_i^S | \mathbf{X}_S, G_S)}[g_j(x_{ij}^S, y_i^S)] \\
\frac{\partial \mathcal{O}(\theta)}{\partial \beta_j} &= \mathbb{E}[g'_j(x_{ij}^T, y_i^T)] - \mathbb{E}_{P_{\beta_j}(y_i^T | \mathbf{X}_T, G_T)}[g'_j(x_{ij}^T, y_i^T)] \\
\frac{\partial \mathcal{O}(\theta)}{\partial \mu_k} &= \mathbb{E}[h_k(Y_c^S) + h_k(Y_c^T)] - \mathbb{E}_{P_{\mu_k}(Y_c | \mathbf{X}_S, \mathbf{X}_T, G_S, G_T)}[h_k(Y_c^S) + h_k(Y_c^T)],
\end{aligned} \tag{13}$$

where we assume that x_{ij} and y_i are uniformly distributed in the given dataset, and thus have a uniform distribution for $P(x_{ij}, y_i)$; $\mathbb{E}[g_j(x_{ij}^S, y_i^S)]$ is the expectation of the local factor function $g_j(x_{ij}^S, y_i^S)$ given the data distribution in the source network and $\mathbb{E}[g'_j(x_{ij}^T, y_i^T)]$ is the expectation of the local factor function $g'_j(x_{ij}^T, y_i^T)$ given the data distribution in the target network; $\mathbb{E}[h_k(Y_c^S) + h_k(Y_c^T)]$ is the expectation of factor function $h_k(Y_c^S) + h_k(Y_c^T)$ given the data distribution (i.e., the average value of the factor function $h_k(Y_c)$ over all triads in the source and the target networks); and the second term in each equation, that is, $\mathbb{E}_{P_{\alpha_j}(y_i^S | \mathbf{X}_S, G_S)}[g_j(x_{ij}^S, y_i^S)]$, $\mathbb{E}_{P_{\alpha_j}(y_i^T | \mathbf{X}_T, G_T)}[g'_j(x_{ij}^T, y_i^T)]$, and $\mathbb{E}_{P_{\mu_k}(Y_c | \mathbf{X}_S, \mathbf{X}_T, G_S, G_T)}[.]$, respectively, represents the expectation under the distribution learned by the model, that is, $P_{\alpha_j}(y_i^S | \mathbf{X}_S, G_S)$, $P_{\alpha_j}(y_i^T | \mathbf{X}_T, G_T)$, $P_{\mu_k}(Y_c | \mathbf{X}_S, \mathbf{X}_T, G_S, G_T)$.

As the graphical structure can be arbitrary and may contain cycles, we use loopy belief propagation (LBP) [Murphy et al. 1999] to approximate the gradients. In order to leverage the unlabeled relationships, we need to perform the LBP process twice in each iteration, one time for estimating the marginal distribution of unknown variables y_i =? and the other time for marginal distribution over all cliques. Finally, with the gradient, we update each parameter with a learning rate η. Regarding the learning rate η, we set its value by adopting an empirical but efficient way—first we use a large η and gradually decrease its value in the following learning iterations. Such a method has been widely used in machine learning. We can also see that in the learning process, the algorithm uses an additional loopy belief propagation to predict the label of unknown relationships. After learning, all unknown relationships are assigned with labels that maximize the marginal probabilities.

Notes. It is worth noting that the social patterns studied in Section 5 capture the common features between the two networks and the proposed TranFG model can be generalized to different networks by incorporating the social patterns, although

there is still one more challenge, namely, network unbalance. The scale of the input two networks may be very unbalanced: one is extremely large and the other is much smaller. This makes the performance (including efficiency and effectiveness) of transfer learning unstable over different networks.

7. ACTIVE TRANSFER LEARNING

To deal with the network unbalance problem, we propose using the active learning method to enhance the proposed TranFG model. With active learning, we aim to minimize the labeled relationships in the target network. As some social patterns have been studied on the labeled relationships, it is necessary to have some sufficient labeled information. As demonstrated in Section 9.2, the amount of labeled relationships in the target network indeed has a strong influence on the prediction performance. With active learning, our goal is to minimize the number of labeled relationships without hurting the prediction performance. We consider several different strategies for active learning, and the experimental results demonstrate the effectiveness of the proposed strategies.

There are a number of active learning methods such as maximum uncertainty and information density [Settles and Craven 2008], while most methods do not consider the network information. In this section, we first introduce two basic methods, Maximum Uncertainty (MU) and Maximum Representativity (MR), which do not consider the network information. Then we present a Maximum Model Influence (MMI) method, which actively selects unlabeled relationships to query by considering both the relationship's uncertainties and the network information. The MMI method was first presented in Zhuang et al. [2012].

Maximum Uncertainty (MU). The straightforward strategy for active learning is to select the most uncertain instances (i.e., relationships in our case) in the target network. The uncertainty of an unlabeled relationship y_i is measured by its *entropy* $H(y_i)$:

$$MU(y_i) = H(y_i) = -\sum_{y \in \mathcal{Y}} P(y_i = y) \log P(y_i = y), \tag{14}$$

where $P(y_y = y)$ can be obtained after learning the factor graph model. For each relationship, we calculate its entropy and then rank all relationships according to the obtained entropy scores. Finally, in the active learning, we select m relationships with the highest entropy scores.

Maximum Representativity (MR). The Maximum Uncertainty strategy tends to choose outliers. One idea to avoid this is to use the strategy of Information Density [Settles and Craven 2008], with a goal of choosing the most *representative* (unlabeled) relationships, which are supposed to be the most informative ones. In particular, we can measure the informativeness of a relationship by its cosine similarity to all other unlabeled relationships in the sense of the attributes associated with the relationship. The informativeness of a relationship can be defined as

$$MR(y_i) = H(y_i) \times \left[\frac{1}{|Y^U|} \sum_{j \in Y^U} \text{sim}(\mathbf{x}_i, \mathbf{x}_j) \right], \tag{15}$$

where $\text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \times \|\mathbf{x}_j\|}$. Again, in the active learning, we select m relationships with the highest representative scores.

Maximum Model Influence (MMI). In the proposed TranFG model, relationships are correlated with each other, and awareness of some "influential" relationships may

help predict the type of the other relationships. However, the previous two strategies do not consider the correlation information. We present an influence propagation method based on the idea from the Linear Threshold Model (LTM) in Kempe et al. [2003]. The LTM model is an influence maximization model, aiming to find a subset of nodes (seed nodes) in a network that could maximize the spread of influence. The LTM model sets a threshold value ε_i for each node and a weight $b_{i,j}$ for its edge between nodes i and j, satisfying $\sum_{j \in NB(i)} b_{i,j} \leq 1$, where $NB(i)$ denotes a set of neighbors of node i. In each timestamp, if $\sum_{j \in NB(i) \wedge activated(j)} b_{i,j} \geq \varepsilon_i$, then the node i will be activated.

We develop a variation of the LTM by incorporating a score for each relationship reflecting the strength of the influence spreading in our model. The basic idea of the active learning method with Maximum Model Influence is that if we actively label a set of unknown relationships and this triggers many other relationships gaining a score so that each of their scores is larger than its threshold, then we say this is a good choice. To quantify this, we define the following propagation process: (1) **Initialization:** the graph is the same as the TranFG model. We call a relationship "activated" when its label y_i is given. The initial activated set of relationships is the set of labeled relationships Y^L. We assign a threshold $\varepsilon_i = \sum_{y \in \mathcal{Y}} |P(y_i = y|G, Y^L) - \frac{1}{|\mathcal{Y}|}|$ for each relationship. In this sense, a relationship with higher uncertainty will be easier to activate. (2) **Influence:** when a relationship e_i is activated, it spreads its gained score increment $(g_i - \varepsilon_i)$ to its neighbor relationships $e_j \in NB(i)$ in the factor graph with a weight $b_{i,j}$, that is, $g_j \leftarrow g_j + b_{i,j}(g_i - \varepsilon_i)$.[10] (3) **Spread:** if a relationship is labeled by the user, we set it as activated and assign its gained score as 1. The gained score for other relationships is set to 0 at the beginning. Once an inactivated relationship e_k gains a score that exceeds the threshold, that is, $g_k > \varepsilon_k$, it will become activated and spread its gained score similarly. Finally, the Maximum Model Influence score $MMI(y_i)$ is defined as the total number of activated relationships in the target network after the propagation process when we have the relationship label y_i for an unknown relationship e_i. In our experiments, to efficiently evaluate the influence of a relationship in MMI, we employ a similar method as that developed in Chen et al. [2009].

Algorithm and Analysis. Finding a set of m relationships that maximizes the total of the Maximum Model Influence scores is NP-hard as proved in Kempe et al. [2003]. Thus, similarly, we use a greedy strategy to approximate the solution with a ratio $(1 - 1/e)$ of the optimal solution. Specifically, in the active learning process, we calculate the MMI score for each unlabeled relationship and finally select the relationship with the highest representative score in each iteration. We give an analysis for the approximation ratio of the greedy algorithm as follows.

To begin with, we first give the definition of the submodular set function.

Definition 7.1 (*Submodular*). A set function F defined on set S is called submodular if for all $A \subset B \subset S$ and $s \notin B$, it satisfies

$$F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B).$$

Function F is monotone increasing if for all sets $S \subseteq T \subseteq V$, there is

$$F(T) \geq F(S).$$

For a function F that is both monotonically increasing and submodular, we could actively select k relationships one by one into a set T. Suppose the relationships are $y_1, y_2, \ldots, y_k$, and we use T_i to denote the set of the ith step $\{y_1, y_2, \ldots, y_i\}$ $(1 \leq i \leq k)$.

[10]The gained score increment reflects the improvement of confidence brought by active user labeling. We empirically set the weight $b_{i,j} = 1/|NB(j)|$.

We see that each time we add a relationship into T, there is an increment of $F(T)$. If some of the previous relationships were not added, this increment becomes larger or stays the same. At each step, we choose to add a relationship $y \in E$ that maximizes $F(T \cup \{y\}) - F(T)$. In this way, we can use a greedy heuristic, that is, each time we choose the relationship that increases $f(T)$ the most, that is,

$$F(T_2) - F(T_1) \geq F(T_3) - F(T_2) \geq \cdots \geq F(T_k) - F(T_{k-1}).$$

Intuitively, the greedy algorithm can generate a good approximate solution for the problem of sampling a k-relationship set T that maximizes $F(T)$. Suppose the generated set is T and the optimal set is T^*. We consider the set $T \cup T^*$, whose function value is larger than (or worse case equal to) that of T^* according to the monotonic property of F. We can construct $T \cup T^*$ as follows: we first use a greedy heuristic and add k relationships of T into it, and then we add relationships in $T^* - T$ one by one. We see that for every relationship in $T^* - T$, the increment is not larger than the increment of any relationship in T. Since $l \leq k$, there must be $2 \cdot F(T) \geq F(T \cup T^*) \geq F(T^*)$. A tighter bound is reported in Nemhauser et al. [1978]. For every monotonically increasing, submodular, nonnegative function F on E, the set generated by the greedy heuristic is at least $(1 - 1/e)$ of the optimal solution.

8. DISTRIBUTED LEARNING

As real social networks may contain millions of users and relationships, it is important for the learning algorithm to scale up well with large networks. To address this, we develop a distributed learning method based on MPI (Message Passing Interface). The scalable performance of the proposed method will be presented in Section 9.

Basically, the learning algorithm consists of two steps: (1) compute the gradient for each parameter via loopy belief propagation and (2) optimize all parameters with the gradient descents. The most expensive part is the step of calculating the gradient. Therefore, we develop a distributed algorithm to speed up the process. We adopt a *master-slave* architecture; that is, one master machine is responsible for optimizing parameters, and the other slave machines are responsible for calculating gradients. At the beginning of the algorithm, the graphical model of TranFG is partitioned into M roughly equal parts, where M is the number of slave processors. This process is accomplished by graph partition software METIS [Karypis and Kumar 1998]. The subgraphs are then distributed over slave nodes. Note that in our implementation, the correlation (factors) between different subgraphs are eliminated, which results in an approximate solution. METIS is a graph partitioning tool. It aims to minimize the cut of edges on a graph such that after the cut, all nodes on the graph can be grouped into M subgraphs. By minimizing the cut, we could obtain a better approximation. In each iteration of the distributed learning, the master machine sends the newest parameters θ to all slaves. Slave machines then start to perform Loopy Belief Propagation on the corresponding subgraph to calculate the "local" belief (the marginal probabilities). Specifically, each slave processor calculates the "local" belief (the marginal probability) on the subgraph G_l according to the following equations (again we use $P(y_i|G, A)$ as the example in the explanation):

$$m_{ij}^l(y_i) = \sigma \sum_{y_i} \psi_{ij}^l(y_i, y_j)\psi_i^l(y_i) \prod_{k \in NB(i)\backslash j} m_{ki}^l(y_i) \tag{16}$$

$$b_i^l(y_i) = \psi_i^l(y_i) \prod_{k \in NB(i)} m_{ki}^l(y_i) \tag{17}$$

$$P(y_i|.) = \sigma \sum_{l=1}^{M} b_i^l(y_i), \tag{18}$$

Table II. Data Transferred in Distributed Learning Algorithm

Phase	From	To	Data Description
Initialization	Master	Slave i	ith subgraph
Iteration Beginning	Master	Slave i	Current parameters θ
Iteration Ending	Slave i	Master	Gradient in ith subgraph

where σ denotes a normalization constant; $m_{ij}^{l}(y_i)$ is the "belief" propagated from node y_j to node y_i; $NB(i)\backslash j$ denotes all nodes neighboring node y_i in the subgraph G_l, except y_j; $\psi_i^l(y_i)$ denotes all defined factor functions related to y_i in the subgraph G_l and is calculated by $\psi_i^l(y_i) = \exp(\sum_{j=1}^{d} \alpha g(y_i^S, x_{ij}^S) + \sum_{j=1}^{d} \beta_j g'(y_i^T, x_{ij}^T))$; $\psi_i^l(y_i)$ denotes all correlation factor functions related to y_i in the subgraph; and notation $b_i^l(y_i)$ denotes the unnormalized "local" belief collected from each subgraph.

After obtaining the local belief, each slave computes the parameter gradient and sends it back to the master. Finally, the master machine collects and sums up all gradients obtained from different subgraphs and updates parameters by the gradient descent method. The data transferred between the master and slave machines are summarized in Table II.

There is one more thing worth noting. As many correlation factors are defined over triads, if we simply eliminate those correlation factors that are defined across different subgraphs, the eliminated triad-based factors would take a large portion of the totally defined triad-based factors, even with a small number of M. This significantly hurts the performance of the proposed TranFG model. To alleviate this problem, we present a virtual-node-based method. In particular, suppose three relationships (y_1, y_2, y_3) in the graphical model are associated with a triad factor $h(y_1, y_2, y_3)$. If the partition assigns two relationships (e.g., y_1 and y_2) into one subgraph and the other one (i.e., y_3) into another subgraph, then we create a virtual relationship in the first subgraph so that the triad factor can still be calculated in the subgraph. (We do not consider the local attribute factor associated with the virtual relationship to avoid redundancy.) If the three relationships are assigned to three different subgraphs, we then randomly select a subgraph and create two virtual relationships to calculate the triad-based correlation factor function and ignore its computation in the other two subgraphs.

9. EXPERIMENTAL RESULTS

9.1. Experimental Setup

The proposed framework is very general and can be applied to many different networks. For experiments, we consider six different networks: Epinions, Slashdot, MobileU, MobileD, Coauthor, and Enron. On the first three networks (Epinions, Slashdot, and MobileU), our goal is to predict undirected relationships (e.g., friendships), while on the other three networks (MobileD, Coauthor, and Enron), the goal is to predict directed relationships (e.g., advisor-advisee relationships).

Comparison Methods. We compare the following methods for predicting the type of social relationships.

SVM: Similar to the logistic regression model used in Leskovec et al. [2010a], SVM uses attributes associated with each relationship as features to train a classification model and then employs the classification model to predict relationships' labels in the test dataset. For SVM, we employ SVM-light.

CRF: It trains a conditional random field [Lafferty et al. 2001] with attributes associated with each relationship and correlations between relationships.

PFG: The method is also based on CRF, but it employs the unlabeled data to help learn the predictive model. The method is proposed in Tang et al. [2011].

COCC: It uses coclustering to transfer the labeled information from one network to another network [Dai et al. 2007a]. This is a transfer-learning-based method.

TranFG: It is the proposed approach, which leverages the labeled information from the source network to help predict the type of relationship in the target network.

We also compare with the method TPFG proposed in Wang et al. [2010] for mining advisor-advisee relationships in the publication network. This method is domain specific and thus we only compare with it on the Coauthor network.

In all experiments, we use the same feature definitions for all methods. On the Coauthor network, we do not consider some domain-specific correlation features.[11]

Evaluation Measures. To quantitatively evaluate the proposed model, we consider the following performance metrics:

—**Prediction accuracy.** We apply the learned model by different methods to predict the types of links in the target networks and evaluate its performance in terms of Precision, Recall, and F1-Measure.
—**How social theory can help.** We analyze how social theories can help improve the prediction performance.
—**How active learning can help.** We use different active learning algorithms to select relationships to actively query their labels and evaluate how active learning can help improve the prediction performance.
—**Efficiency and scalability performance.** We evaluate the computational time as the efficiency metric to evaluate the efficiency and scalability performance of the proposed model.

9.2. Prediction Performance and Analysis

Prediction Accuracy Across Heterogeneous Networks. We compare the performance of the four methods for predicting friendships (or trustful relationships) on four pairs of networks: Epinions (S) to Slashdot (T), Slashdot (S) to Epinions (T), Epinions (S) to MobileU (T), and Slashdot (S) to MobileU (T). In all experiments, we use 40% of the labeled data in the target network for training and the rest for test. For transfer, we consider the labeled information in the source network. Table III lists the performance of the different methods on the four test cases. Our approach shows better performance than the three alternative methods. We conducted sign tests for each result, which shows that all the improvements of our approach TranFG over the three methods are statistically significant ($p \ll 0.01$).

Table IV shows the performance of the five methods (including TPFG for mining advisor-advisee relationships on the Coauthor dataset [Wang et al. 2010]) for predicting directed relationships (the source end has a higher social status than the target end) on six pairs of networks: Coauthor (or MobileD) (S) to Enron (T), Enron (or MobileD) (S) to Coauthor (T), and Coauthor (or Enron) (S) to MobileD (T). In each test case, we again use 40% of the labeled data in the target network for training and the rest for test, while for transfer we consider the labeled information from the source network. We see that by leveraging the supervised information from the source network, our method clearly improves the performance (about 15% by F1-score on Coauthor, 20% on MobileD, and 23% on Enron). Another phenomenon is that the prediction accuracy is correlated with the size of the source network. For example, when predicting the manager-subordinate relationship in the Enron network, we obtain a much higher

[11]We conducted experiments but found that those features will lead to overfitting.

Table III. Performance Comparison of Different Methods for Predicting Friendships (or Trustful Relationships)

Dataset	Method	Prec.	Rec.	F1-Score
Slashdot (T) (40%)	SVM	0.7157	**0.9733**	0.8249
	CRF	0.8919	0.6710	0.7658
	PFG	0.9300	0.6436	0.7607
Epinions to Slashdot	COCC	0.8291	0.5511	0.6621
	TranFG	**0.9414**	0.9446	**0.9430**
Epinions (T) (40%)	SVM	0.9132	**0.9925**	0.9512
	CRF	0.8923	0.9911	0.9393
	PFG	**0.9954**	0.9787	**0.9870**
Slashdot to Epinions	COCC	0.9667	0.6732	0.7937
	TranFG	**0.9954**	0.9787	**0.9870**
MobileU (T) (40%)	SVM	0.8983	0.5955	0.7162
	CRF	0.9455	0.5417	0.6887
	PFG	**1.0000**	0.5924	0.7440
Epinions to MobileU	COCC	0.7952	0.6875	0.7374
	TranFG	0.8239	**0.8344**	**0.8291**
MobileU (T) (40%)	SVM	0.8983	0.5955	0.7162
	CRF	0.9455	0.5417	0.6887
	PFG	**1.0000**	0.5924	0.7440
Slashdot to MobileU	COCC	0.8615	0.6022	0.7089
	TranFG	0.7258	**0.8599**	**0.7872**

(S) indicates the source network and (T) the target network. For the target network, we use 40% of the labeled data in training and the rest for test.

Table IV. Performance Comparison of Different Methods for Predicting Directed Relationships (the Source End Has a Higher Social Status Than the Target End)

Dataset	Method	Prec.	Rec.	F1-Score
Enron (T) (40%)	SVM	**0.9524**	0.5556	0.7018
	CRF	0.7778	0.7673	0.7725
	PFG	0.9130	0.7241	0.8077
MobileD to Enron	COCC	0.7647	0.6190	0.6842
Coauthor to Enron	COCC	0.7619	0.6957	0.7273
MobileD to Enron	TranFG (M)	0.8438	0.7941	0.8182
Coauthor to Enron	TranFG (C)	0.9091	**0.8824**	**0.8955**
Coauthor (T) (40%)	SVM	0.6910	0.3727	0.4842
	CRF	**0.8472**	0.2937	0.4362
	PFG	0.8189	0.3377	0.4782
	TPFG	0.5936	**0.7611**	0.6669
MobileD to Coauthor	COCC	0.6614	0.3834	0.4854
Enron to Coauthor	COCC	0.6667	0.3901	0.4922
MobileD to Coauthor	TranFG (M)	0.8235	0.3889	0.5283
Enron to Coauthor	TranFG (E)	0.8193	0.6415	**0.7196**
MobileD (T) (40%)	SVM	0.5249	0.3725	0.4358
	CRF	0.4454	**0.5763**	0.5025
	PFG	**0.8739**	0.3731	0.5229
Enron to MobileD	COCC	0.7152	0.2443	0.3642
Coauthor to MobileD	COCC	0.6412	0.2422	0.3516
Enron to MobileD	TranFG (E)	0.8013	0.4808	0.6010
Coauthor to MobileD	TranFG (C)	0.8323	0.5154	**0.6366**

(S) indicates the source network and (T) the target network. For the target network, we use 40% of labeled data in training and the rest for test.

Table V. Prediction Accuracy between Homogeneous Networks

Dataset	Method	Prec.	Rec.	F1-Score
Slashdot (S) to Slashdot (T) (40%)	PFG	0.9300	0.6436	0.7607
	TranFG	0.9948	0.9185	0.9551
	TranFG-Heter	0.9414	0.9446	0.9430
Epinions (S) to Epinions (T) (40%)	PFG	0.9954	0.9787	0.9870
	TranFG	0.9954	1.0000	0.9977
	TranFG-Heter	0.9954	0.9787	0.9870
MobileU (S) to MobileU (T) (40%)	PFG	1.0000	0.5924	0.7440
	TranFG	0.9259	0.7895	0.8523
	TranFG-Heter	0.8239	0.8344	0.8291
Enron (S) to Enron (T) (40%)	PFG	0.9130	0.7241	0.8077
	TranFG	0.9394	0.9688	0.9538
	TranFG-Heter	0.9091	0.8824	0.8955
Coauthor (S) to Coauthor (T) (40%)	PFG	0.8189	0.3377	0.4782
	TranFG	0.8321	0.7433	0.7852
	TranFG-Heter	0.8193	0.6415	0.7196
MobileD (S) to MobileD (T) (40%)	PFG	0.8739	0.3731	0.5229
	TranFG	0.8843	0.6115	0.7230
	TranFG-Heter	0.8323	0.5154	0.6366

TranFG-Heter is the best performance obtained in the heterogeneous transfer (Cf. Tables III and IV).

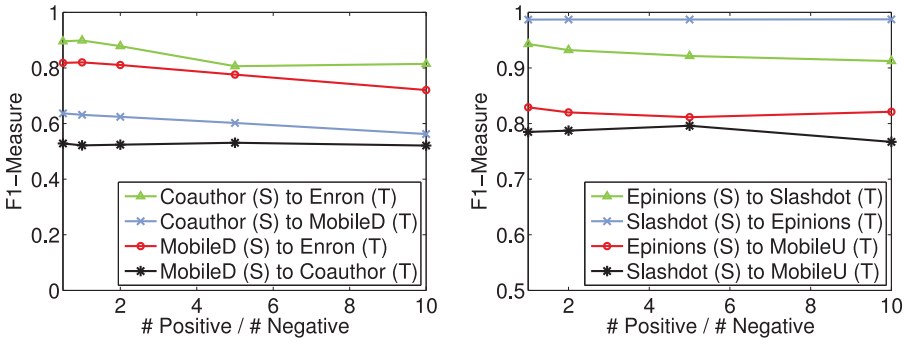
accuracy by using the Coauthor network (1,310 authors and 6,096 relationships) as the source network than that of using the MobileD network (232 users and 3,567 relationships) as the source network (89.6% vs. 81.8%). From the results, we can also see that the transferring performance is different for different prediction tasks. For example, transferring from Epinions is better than transferring from Slashdot for the MobileU network. In practice, the performance of transfer link prediction depends on how the source network is correlated with the target network. If the two networks are more similar to each other, then more information can be transferred from the source network to help the target network.

The method PFG can be viewed as a nontransferable counterpart of our method, which does not consider the labeled information from the source network. From both Tables III and IV, we can see that with the transferred information, our method clearly improves the relationship categorization performance. Another phenomenon is that PFG has a better performance than the other two methods (SVM and CRF) in most cases. PFG leverages the unlabeled information in the target network, and thus enhances the performance. The only exception is the case of Epinions (S) to Slashdot (T), where it seems that users in Slashdot have a relatively consistent pattern, and merely with some general features such as in-degree, out-degree, and number of common neighbors, a classification-based method (SVM) can achieve very high performance.

Prediction Accuracy Across Homogeneous Networks. We study how well the algorithm performs when the source network and the target network are homogeneous, that is, the same type of networks. In particular, we partition each of the six networks into two subnetworks and use one of them as the source network and the other as the target network. For the target network, again we consider 40% of the labeled information, and for the source network, we consider all the labeled information.

Table V shows the accuracy of predicting relationships by learning across homogeneous networks. The homogeneous transfer actually represents the upper bound for heterogeneous transfer. We see that our model for heterogeneous transfer (TranFG-Heter) performs close to the upper bound.

Bias Analysis for Transfer Link Prediction. From Table I, we see that for some networks, such as Epinions and Slashdot, the number of positive relationships (trust



(a) Directed relationships (b) Friendships (or trust relationships)

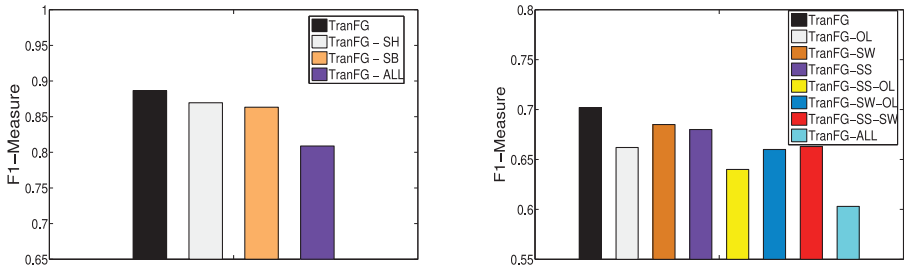
Fig. 12. Bias analysis for transfer link prediction by the proposed TranFG.

and friend relationships) is much greater than that of negative relationships, while in some other networks, such as Coauthor and Enron, the situation is completely different. We conducted an experiment to study how the unbalanced positive/negative relationships influence the performance of transfer link prediction. Figure 12 shows the results of transfer link prediction for directed relationships and undirected (friend and trust) relationships by varying the ratio of positive/negative instances in the source network. For example, from Coauthor to Enron, #positive/#negative=2 indicates that the number of positive relationships is twice the number of negative relationships in the Coauthor network. Specifically, we fixed the number of positive relationships in Coauthor and randomly sampled half of the number of negative relationships. We see that the prediction performance is not sensitive to the bias problem. This confirms the effectiveness of the proposed TranFG model.

9.3. How Can Social Theories Help?

We now analyze how different social theories (social balance, social status, structural hole, two-step flow (opinion leader), and strong/weak tie) can help predict social ties. From Section 5, we see that the social theory has a similar pattern on different networks, which can be leveraged to bridge the different networks. At the same time, from Figures 5 to 10, we can also see there is still a difference between different networks. For example, in Figure 6, the obtained probabilities on MobileU are different from those obtained from the other two networks. In the transfer learning, the TranFG model will determine the weight of different social-theory-based factors for different learning tasks. For example, in learning to predict social ties from Coauthor to MobileD, the learned weights of the four social-balance-based factors are, respectively, 0.002248, 0.000121, −0.001544, and −0.000825, which indicates that the first two balanced structure-based factors have a positive correlation between the two networks, while the other two unbalanced structure-based factors have a negative correlation between the two networks.

To predict friendships, we consider social balance (SB)- and structural hole (SH)-based transfer, and to predict directed friendships, we consider social status (SS)-, opinion leader (OL)-, and strong/weak tie (SW)-based transfer. Here we examine the contribution of the different factors defined in our TranFG model. Figure 13 shows the average F1-Measure score over the different networks, obtained by the TranFG model for predicting friendships and directed relationships. In particular, TranFG-SB represents that we remove social-balance-based transfer features from our model; TranFG-SW-OL represents that we remove both strong/weak-tie- and



(a) Friendship (b) Directed

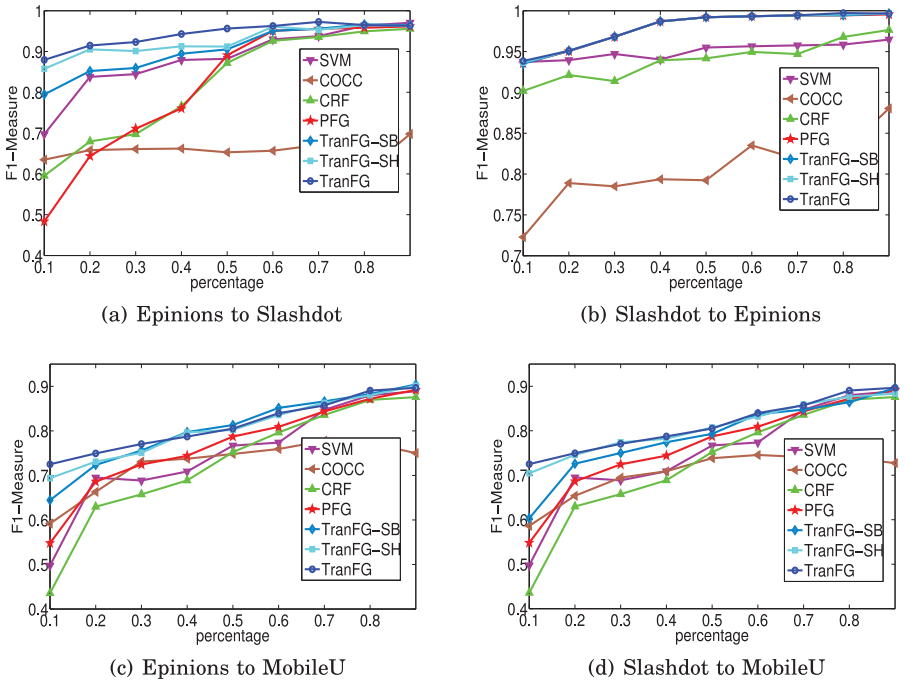
Fig. 13. Factor contribution analysis. TranFG-SH denotes our TranFG model by ignoring the structural-hole-based transfer. TranFG-SB stands for ignoring the structural-balance-based transfer. TranFG-OL stands for ignoring the opinion-leader-based transfer, and TranFG-SS stands for ignoring social-status-based transfer.

opinion-leader-based transfer features; and TranFG-All denotes that we remove all the transfer features. It can be clearly observed that the performance drops when ignoring each of the factors. We can also see that, for predicting friendships, the social balance is a bit more useful than structural hole, and for predicting directed relationships, the opinion leader factor is more important than the factors of strong/weak tie and structural hole. The analysis confirms that our method works well (further improvement is obtained) by combining different social theories.

Social-Balance- and Structural-Hole-Based Transfer. We present an in-depth analysis on how the social-balance- and structural-hole-based transfer can help by varying the percent of labeled training data in the target network, as shown in Figure 14. We see that in all cases except Slashdot to Epinions, clear improvements can be obtained by using the social-balance- and structural-hole-based transfer, when the labeled data in the target network is limited ($\leq$50%). Indeed, in some cases such as Epinions to Slashdot, with merely 10% of the labeled relationships in Slashdot, our method can obtain a good performance (88% by F1-score). Without transfer, the best performance is only 70% (obtained by SVM). We also find that structural-balance-based transfer is more helpful than structural-hole-based transfer for predicting friendships in most cases with various percentages of labeled relationships. This result is consistent with that obtained in the factor contribution analysis.

A different phenomenon is found in the case of Slashdot to Epinions, where all methods can obtain an F1-score of 94% with only 10% of the labeled data. The knowledge transfer seems not helpful. After a thorough investigation, we found that a high accuracy (about 90%) could be achieved simply using those features (Cf. appendix for details) defined on the relationships. The structure information indeed helps, but the gained improvement is limited.

Social-Status-, Opinion-Leader-, and Strong/Weak-Tie-Based Transfer. Figure 15 shows an analysis for predicting directed relationships on the six cases (Coauthor to Enron, MobileD to Enron, Enron to Coauthor, MobileD to Coauthor, Coauthor to MobileD, and Enron to MobileD). Here, we focus on testing how social-status-, opinion-leader-, and strong-tie-based transfer can help predict the types of relationships by varying the percentage of labeled relationships in the target network. In almost all cases, the TranFG model achieves consistent improvements. The only exception is the case of MobileD to Coauthor, in which, when the labeled data in the target network is more than 80%, SVM works the best in all the methods. There may be two reasons: the first is that the MobileD network is small, which limits the ability of transfer, and



(a) Epinions to Slashdot (b) Slashdot to Epinions

(c) Epinions to MobileU (d) Slashdot to MobileU

Fig. 14. Performance of predicting friendships with and without the balance- and structural-hole-based transfer by varying the percentage of labeled data in the target network.

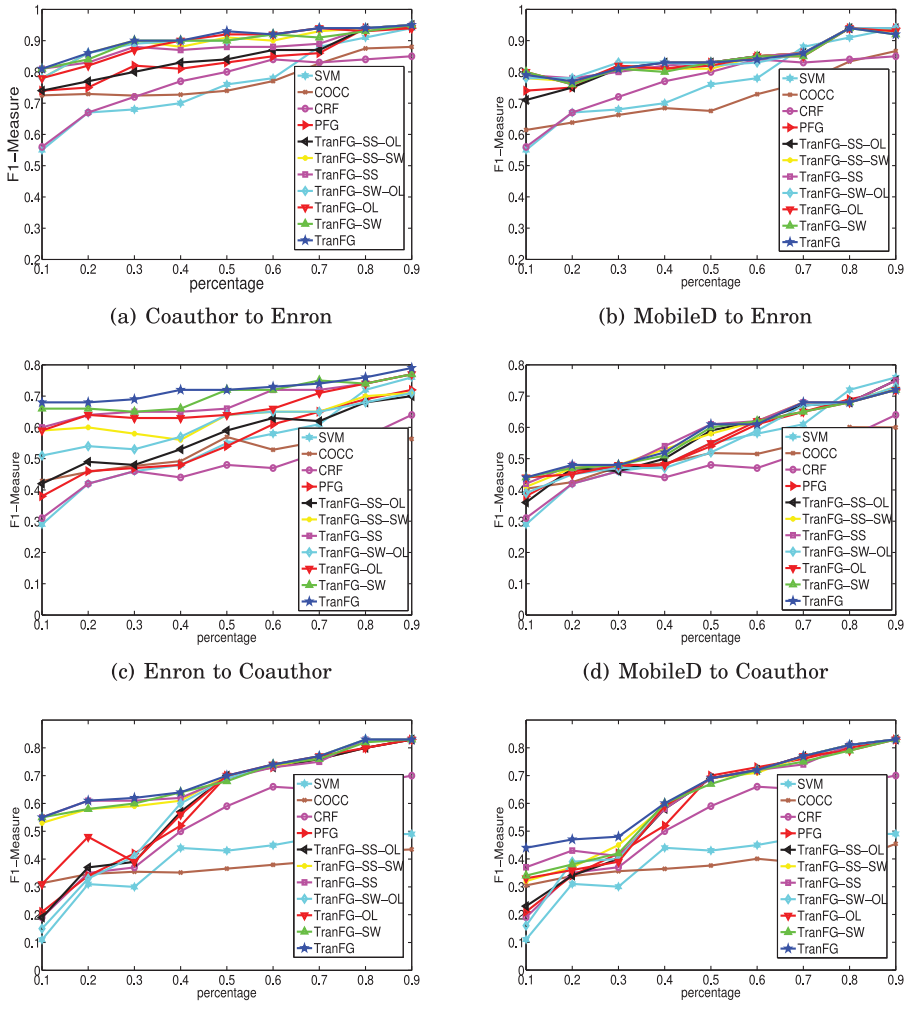
the second reason is that the large proportion of the labeled data in the target network would provide statistically sufficient data for SVM to learn a good classification model. In all the other cases the social-theory-based transfer indeed helps. For example, when there are only 10% of labeled advisor-advisee relationships in the Coauthor network, and the social theory (status, opinion leader, and strong/weak tie)-based transfer is not considered, the F1-score is only 30%. By leveraging the status- and opinion-leader-based transfer from the email network (Enron), the score is more than doubled (+60%). With the transfer from the mobile network (MobileD), the prediction accuracy is also significantly improved (+50%). Moreover, we find that the opinion-leader-based transfer is more helpful than the social-status-based transfer with various percentages of the labeled data, while both opinion-leader- and social-status-based transfer are more useful than strong/weak-tie-based transfer.

9.4. Active Learning Performance

For active learning, in each experiment, we use different active learning algorithms to select a proportion τ of relationships in the target network to query their labels. We range the proportion τ from 5% to 50% with an interval of 5%. After each round of selection, we apply the proposed TranFG model to learn and predict the type of relationships in the target network. We implement the experiment 10 times on each dataset and report the average F1-score.

We compare the following active learning algorithms:

—**Random:** It randomly selects the relationships in the target network to query.
—**Maximum Uncertainty (MU):** It uses the maximum uncertainty method to select the most uncertain relationships in the target network to query.



(a) Coauthor to Enron

(b) MobileD to Enron

(c) Enron to Coauthor

(d) MobileD to Coauthor

(e) Coauthor to MobileD

(f) Enron to MobileD

Fig. 15. Performance of predicting directed relationship with and without the status-, opinion-leader-, and strong/weak-tie-based transfer by varying the percentage of labeled data in the target network.

—**Maximum Representativity (MR):** It uses the maximum representativity method to select the relationships in the target networks to query.
—**Maximum Model Influence (MMI):** It considers both attributes associated with each relationship and the network information to actively select the relationships in the target network to query.

Figure 2 shows the active learning performance by the different algorithms on four tasks for predicting social ties across different networks. It can be clearly seen that the active learning algorithms significantly improve the performance over the random method. For example, for predicting social ties from Epinions to Slashdot, with the active learning by the MMI strategy, the performance of using merely 15% of the labeled relationships in the target network (Slashdot) is already close to the performance of the random strategy using 30% of the labeled relationships in the target network.

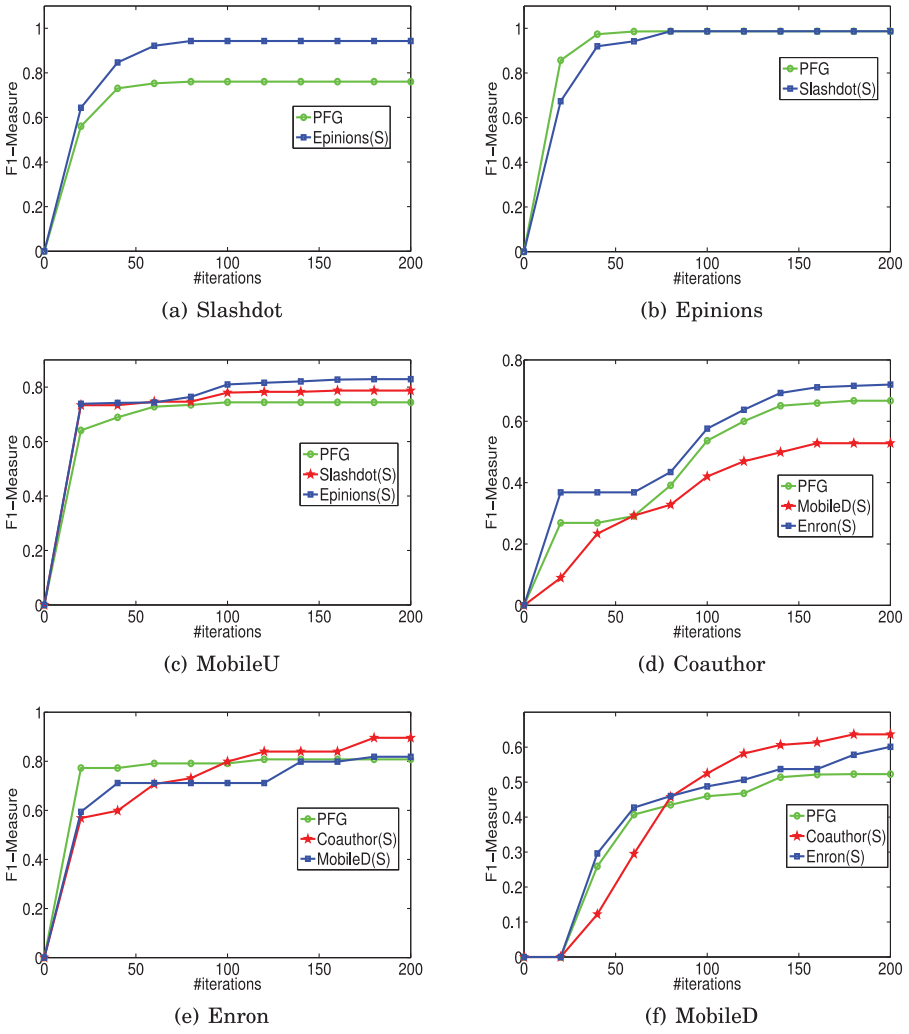


Fig. 16. Convergence analysis of learning algorithm on six different networks.

From the results, we can also see that the network information indeed helps. The MMI strategy incorporates both attributes associated with each relationship and the network information. On average, it reaches a better performance (+2%) than the other two strategies (MU and MR) that do not consider the network information.

9.5. Efficiency and Scalability Performance

We first evaluate the convergence of the learning algorithm for the proposed model and then test its efficiency and scalability.

Convergence Analysis. We conduct an experiment on the effect of the number of iterations of the loopy belief propagation. Figure 16 shows the convergence analysis results of the learning algorithm. We see on all the test cases that the learning algorithm converges in fewer than 100 iterations. In some cases (such as Slashdot to MobileU and

Table VI. Efficient Performance When Training TranFG on Different Networks (Minute)

Dataset	None	MobileD (T)	Coauthor (T)	Enron (T)
MobileD(S)	0.77	1.40	0.96	0.69
Coauthor(S)	0.25	1.05	0.40	0.13
Enron(S)	0.01	0.88	0.23	0.01
Dataset	None	MobileD (T)	Coauthor (T)	Enron (T)
Slashdot(S)	16.62	27.83	42.05	9.93
Epinions(S)	23.02	35.59	50.48	16.73
MobileU(S)	0.06	16.61	23.22	0.09

Coauthor to Enron), the performance becomes stable after merely 20 iterations. This suggests that the learning algorithm is efficient and has a good convergence property.

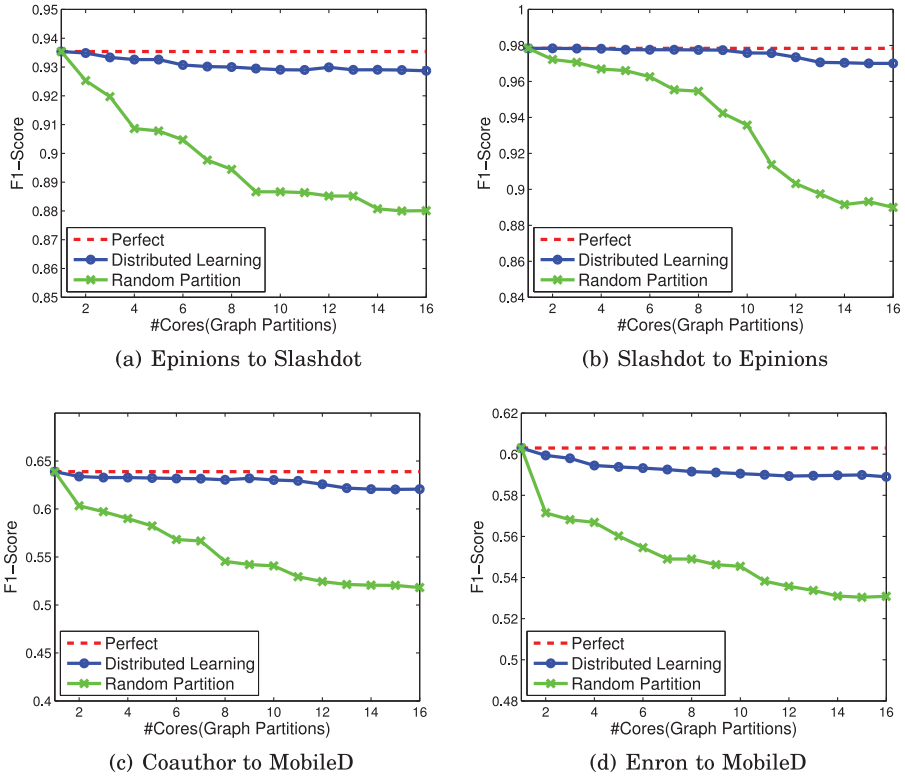
Efficiency. It took about 1 to 45 minutes to train the TranFG model over different datasets (e.g., 42 minutes for learning over the Slashdot and the Epinions networks). The efficiency of the proposed TranFG model is acceptable, as training the TranFG model in most test cases can be completed within a few minutes. The most time-consuming case is to train the transfer model over the Epinions network, which consists of 131,828 nodes and 841,372 relationships. The model learning can still be completed in less than 1 hour. Table VI lists the efficiency performance when training the TranFG model over different networks. For incorporating social balance and social status into the TranFG model, we need to count all triads in the source and the target networks. We design an efficient algorithm that is linear in the number of relationships. The algorithm takes 1 to 12 minutes to enumerate all triads for the six networks. The algorithm is available online.[3]

Scalability. Figure 3 shows the scalability performance of the distributed learning algorithm for the TranFG model. It gives the running time and speedup of the distributed algorithm with different numbers of computer nodes (two, four, six, eight, 10, 12, 14, 16 cores) used. The speedup curve is close to the perfect line at the beginning. Although the speedup inevitably decreases when the number of cores increases due to the communication cost, on average, it can achieve a $\sim 9\times$ speedup with 12 cores. Please also note that the speedup performance depends on the structure of the source and the target networks. For example, the speedup performance of Slashdot to Epinions is better than that of Coauthor to MobileD. This is because the speedup performance depends on the graph partition, which is closely related to the network structure.

We further verify whether the distributed learning will hurt the prediction accuracy. Figure 17 shows how the distributed learning influences the prediction accuracy performance (F1-score). We see that the performance of the TranFG model with distributed learning only decreases slightly compared with the performance (referred to as "Perfect" in Figure 17) of model learning on a single machine. With distributed learning with the TranFG model on 16 cores, on average the accuracy only drops 0.8% in terms of F1-score. We also evaluate the performance of distributed learning by randomly partitioning the graph into M subgraphs. Comparing with our method, which partitions the graph by METIS (according to the theory of min cut), the random strategy results in a drop (about a 10% drop when learning on 16 cores) in the prediction accuracy. This confirms the effectiveness of the design of the distributed learning.

10. RELATED WORK

Inferring Social Ties. Inferring social ties is an important problem in social network analysis. Liben-Nowell et al. [2007] presented an unsupervised method for link prediction. They studied different algorithms and found that the Katz algorithm



(a) Epinions to Slashdot (b) Slashdot to Epinions

(c) Coauthor to MobileD (d) Enron to MobileD

Fig. 17. Prediction accuracy (F1-score) by the distributed learning. The red dashed line is the accuracy performance on a single machine.

can achieve the best performance. Xiang et al. [2010] developed a latent variable model to estimate relationship strength from interaction activity and user similarity. Backstrom et al. [2011] proposed a supervised random walk algorithm to estimate the strength of social relationships. Leskovec et al. [2010a] employed a logistic regression model to predict positive and negative relationships in online social networks. Hopcroft et al. [2011] studied the extent to which the formation of a reciprocal relationship can be predicted in a dynamic network. However, most existing works focus on predicting and recommending unknown relationships in social networks but ignore the types of relationships.

There are several works on inferring the meanings of social relationships. Diehl et al. [2007] tried to identify the manager-subordinate relationships by learning a ranking function. Wang et al. [2010] proposed an unsupervised probabilistic model for mining the advisor-advisee relationships from the publication network. Crandall et al. [2010] investigated the problem of inferring friendship between people from co-occurrence in time and space. Eagle et al. [2009] presented several patterns discovered in mobile phone data and tried to use these patterns to infer the friendship network. However, these algorithms mainly focus on specific domains, while our model is general and can be applied to different domains. More importantly, our work takes the first step to incorporate social theories for inferring social ties across heterogeneous networks. Zhuang et al. [2012] presented a learning framework for inferring social ties in a single network and also propose an active learning method. Our presented active learning algorithm is similar to that in Zhuang et al. [2012]. The major difference is that we

consider this problem in two different networks, while Zhuang et al. [2012] considered the active learning problem in a single network. In our previous work [Tang et al. 2012], we studied the problem of inferring social ties across heterogeneous networks. In this work, we make the extension from the following aspects. First, we further investigate how strong/weak ties hypotheses influence the formation of different types of social ties. Second, we give a more detailed definition of the social-theory-based transfer features. Third, we evaluate the proposed model on a new dataset, MobileD, where we aim to infer manager-subordinate relationships between mobile users. Last, we present and compare the accuracy performance of inferring social ties across homogeneous networks and the efficiency performance of the proposed model.

Link Prediction. Our work is related to link prediction, which is one of the core tasks in social networks. Existing works on link prediction can be broadly grouped into two categories based on the learning methods employed: unsupervised link prediction and supervised link prediction. We review representative approaches of each category and highlight the difference between existing works and our effort. Unsupervised link predictions usually assign scores to potential links based on the intuition that the more similar users in a pair are, the more likely they are linked. Various similarity measures of users are considered, such as the Adamic and Adar measure [Adamic and Adar 2001], the preferential attachment [Newman 2001], and the Katz measure [Katz 1953]. A survey of unsupervised link prediction can be found in Liben-Nowell and Kleinberg [2007]. Most of these works study the link prediction task using some macro-level features but do not consider the social effects, such as the social balance effect. Recently, Lichtenwalter et al. [2010] designed a flow-based method for link prediction. There are also a few works that employ supervised approaches to predict links in social networks, such as Backstrom and Leskovec [2011], Dong et al. [2012], and Leskovec et al. [2010a]. Zhang et al. [2013] studied the problem of link prediction for new users across aligned heterogeneous social networks. Dong et al. [2015] studied the problem of link prediction in coupled networks, where the structure information of one (source) network and the interactions between this network and another (target) network is available, and the goal is to predict the missing links in the target network.

The main difference between existing works on link prediction and our effort is that existing works mainly focus on a single network, while our proposed model combines social theories (such as structural balance, structural hole, and social status) into a transfer learning framework and can be applied to different domains.

Social Behavior Analysis. Another type of related work is social behavior analysis. Tan et al. [2010] investigated how social actions evolve in a dynamic social network and propose a time-varying factor graph model for modeling and predicting users' social behaviors. Tang and Liu [2011] developed a framework for classifying the types of social relationships by learning across heterogeneous networks. The types of social relationships are specific and they do not consider learning a general model for inferring social ties across any different networks. Yang et al. [2010] studied the retweeting behavior in the Twitter network. Retweet behaviors between users are very relevant to social relationships. The strong social tie may result in a higher likelihood of retweeting. Tan et al. [2011] investigated how different types of relationships between users influence the change of users' opinion. They found that by incorporating the social relationships, the performance of user-level sentiment analysis can be significantly improved. Zhang et al. [2015] proposed a method named COSNET to connect different networks together, and Yang et al. [2015] proposed a method to find matched entities from different datasets. Dong et al. [2015] tried to infer users' social status enterprise communication networks and study the phenomenon of rich "club." However, all these

works do not consider inferring social ties across multiple different networks. Another related research topic is relational learning [Getoor and Taskar 2007]. Relational learning focuses on the classification problems when objects or entities are presented in relations. A number of supervised methods for link prediction in relational data have also been developed [Taskar et al. 2003]. In this article, we extend the relational learning problem to the transfer learning context and study using social theories to enhance relational learning.

Transfer Learning. Our work is also related to transfer learning, which aims to transfer knowledge from a source domain to a related target domain. Two main issues in transfer learning are "what to transfer" and "when to transfer" [Pan and Yang 2010]. Many approaches have been proposed by selecting instances from the source domain for reuse in the target domain [Dai et al. 2007b; Gao et al. 2008; Liao et al. 2005; Shi et al. 2008]. There are also many works conducted to transfer features between different domains. For example, Argyriou and Evgeniou [2006] proposed a method to learn a shared low-dimensional representation for multiple related tasks. Blitzer et al. [2006] presented a structural correspondence learning (SCL) approach to induce the correspondences among features across two domains. Some other works include to Cao et al. [2010], Jebara [2004], Lee et al. [2007], and Ando and Zhang [2005]. There are a few works about transferring knowledge across heterogeneous feature spaces [Ling et al. 2008]. For example, Dai et al. [2008] proposed translated learning, which can transfer the labeled information across two entirely different domains. Argyriou et al. [2008] proposed an algorithm for the classification problem in the heterogeneous environment. Compared with existing works, this work is different in the following aspects. First, most existing works only consider homogeneous networks (the source and the target network are of the same type), while the networks studied in our problem are quite different and they may even not have any overlapping attribute features. Second, we combine social-theory-based features into a transfer learning framework, while existing methods are mainly concerned with how to find shared attributes across different domains.

Discussions. Generally speaking, the problem addressed in this article is different from traditional research on inferring social ties and link prediction. The major difference lies in that we study the link prediction problem across different networks. Another and more important contribution of our work to this field is that we systematically investigate various social theories (e.g., social balance, social status, structural hole, two-step flow, and strong/weak tie) and design a principled methodology to combine those social theories into a probabilistic graphical model. From the method's perspective, the proposed solution is related to the probabilistic graph model and transfer learning. The proposed framework TranFG itself is developed based on factor graph models [Kschischang et al. 2001], and the learning algorithm is based on the algorithm for partially labeled factor graphs [Tang et al. 2011]. Our contribution is to extend this model to deal with the problem of transfer link prediction. The fundamental challenge is that no common features can be used to transfer knowledge from the source network to the target network, which makes it infeasible to directly apply existing transfer learning methods to this task. TranFG provides an elegant way to bridge two different networks, by leveraging the common properties (based on the studied social theories) of different networks. The learning framework is to guarantee how likely it is that a common property can be trusted to transfer the knowledge. To the best of our knowledge, this is the first attempt to study the problem of transfer link prediction across heterogeneous networks using social theories.

Table VII. Features Defined on Relationship e_{ij} (or (v_i, v_j)) in Epinions/Slashdot [Leskovec et al. 2010a]

Feature	Description
in-degree	$d_{in}(v_i)$, $d_{in}(v_j)$: two features represent the in-degree of node v_i and v_j, respectively.
out-degree	$d_{out}(v_i)$, $d_{out}(v_j)$: two features represent the out-degree of node v_i and v_j, respectively.
total-degree	$d_{in}(v_i) + d_{out}(v_i)$, $d_{in}(v_j) + d_{out}(v_j)$: two features represent the degree of node v_i and v_j, respectively.
common neighbors	The feature represents the total number of common neighbors of v_i and v_j in an undirected sense.

Table VIII. Features Defined on Relationship e_{ij} (or (v_i, v_j)) in MobileU [Tang et al. 2011; Eagle et al. 2009]

Feature	Description (Users v_i and v_j)
total proximity	The feature represents the total number of proximity events between v_i and v_j.
in-role	The dataset specifies some location as working places. The feature represents the number of proximity events at working place in daytime from Monday to Friday.
extra-role	The feature represents the number of proximity events at home or elsewhere at night on weekends.
total communication	The feature represents the number of communication logs between v_i and v_j.
night call ratio	The feature represents the ratio of communication logs occurring in the night (20:00 p.m.–6:00 a.m.) between v_i and v_j.

The dataset records periodic Bluetooth scan information. If one mobile finds another mobile in a discovery scan, then we say there is a proximity event.

11. CONCLUSION AND FUTURE WORK

In this article, we investigate the problem of inferring the type of social relationships across heterogeneous networks. More accurately, we study how to accurately infer social ties in a target network with only few labeled relationships by leveraging information from a source network. We formulate the problem and propose a transfer-based factor graph model (TranFG). The model incorporates social theories into a semisupervised learning framework, which is used to transfer supervised information from the source network to help infer social ties in the target network. To further improve the proposed model, we present several active learning algorithms and a distributed learning algorithm. We evaluate the proposed model on six different networks. We show that the proposed model can significantly improve the performance for inferring social ties across different networks compared with several alternative methods. With the active learning, we can further obtain a significant improvement on the accuracy performance. The learning algorithm for the model can also be easily distributed. For example, with the distributed learning, we can obtain a 9× speedup with 12 cores. Through the observation analysis on six different types of networks, our study also reveals several interesting phenomena.

The proposed framework (TranFG) has many potential applications. For example, recently we have applied the framework to help two companies mine social relationships from mobile data and bank transaction data. From the mobile data, we are trying to infer family and colleague relationships, which can help the mobile company recommend personalized services. From the bank transaction data, we are trying to infer the type of relationship between two bank accounts. This would be very useful to help the bank find new customers. So far, the obtained results are very promising.

The general problem of inferring social ties represents an interesting research direction in social network analysis. There are many potential future directions of this work. First, some other social theories can be further explored and validated for analyzing the formation of different types of social relationships. Next, it is interesting to further study how incrementally learning the proposed model so that we can directly involve online user interactions in the learning process. Another potential issue is to leverage

the results of inferring social ties to help deal with the information overload problem. For example, on Facebook or Twitter, we may follow thousands of friends, but frankly, we cannot maintain all of them [Gladwell 2001]. Who is in our core circles? Based on the results in this work, we can further study how to identify core circles for each social user. It also has many real applications based on the results of social tie analysis. For example, we can use the inferred social ties to help information recommendation in the social network. According to the social influence theory, a user's connections with different social ties would have very different influences on his or her behaviors from different aspects.

APPENDIX: FEATURE DEFINITION

There are two categories of features. The first category includes local features defined for each specific network, and the second includes transfer features defined based on the social theories. Tables VII through XI give a summary of local feature definitions for the six networks. For a more detailed description of the feature definitions, please refer to the literature [Diehl et al. 2007; Eagle et al. 2009; Leskovec et al. 2010a; Tang et al. 2011].

For the transfer features, in Epinions, Slashdot, and Mobile, we define four (real-valued) balance-triad-based features and six (real-valued) structural-hole-based features. In Coauthor and Enron, we define seven (real-valued) social-status-based features (011, 101, 110, 100, 000, 111 and −111) and four (binary) opinion-leader-based features.

Table IX. Features Defined in Relationship e_{ij} (or (v_i, v_j)) in Coauthor (P_i Denotes a Set of Papers Published by Author v_i [Tang et al. 2011])

Feature	Description
paper count	$\|P_i\|$, $\|P_j\|$: two features represent the number of papers published by v_i and v_j, respectively.
paper ratio	$\|P_i\|/\|P_j\|$: the feature represents the ratio of the number of published papers by v_i to the number by v_j.
coauthor ratio	$\|P_i \cap P_j\|/\|P_i\|$, $\|P_i \cap P_j\|/\|P_j\|$: two features represent the ratio of the number of common coauthors between v_i and v_j to the number of coauthors for each of them, respectively.
conference coverage	The feature represents the ratio of the number of common publication venues between v_i and v_j to the number of publication venues by v_j.
first-pub-year diff	The feature represents the difference in year of the first earliest publication of v_i and v_j.

Table X. Features Defined in Relationship e_{ij} (or (v_i, v_j)) and $v_k \notin \{v_i, v_j\}$ in Enron (Email Counts) [Diehl et al. 2007]

Feature	Description
$v_i - v_j$, $v_j - v_i$	Two features respectively represent the number of emails sent from v_i to v_j (or received by v_i from v_j).
$v_i - v_{\neg j}$, $v_j - v_{\neg i}$	Two features respectively represent the number of emails sent from v_i (or v_j) to a user (e.g., v_k) rather than v_j (or v_i).
$v_{\neg j} - v_i$, $v_{\neg i} - v_j$	Two features respectively represent the number of emails that v_i (or v_j) received from a user (e.g., v_k) rather than v_j (or v_i).
$v_k - v_{i.j}$	The feature represents the number of emails that v_i and v_j received from a common user v_k together.

Table XI. Features Defined in Relationship e_{ij} (or (v_i, v_j)) in MobileD
The dataset is a mobile network of enterprise, in which we try to infer manager-subordinate relationships between users.

Feature	Description
total communication	The feature represents the number of communication logs between v_i and v_j.
call duration	The feature represents the sum of the length of all calls between v_i and v_j.
night call ratio	The feature represents the ratio of the number of communication logs in the night (20:00 p.m.–6:00 a.m.) between v_i and v_j to the total number of calls made between them.
weekend call ratio	The feature represents the ratio of the number of communication logs between v_i and v_j made in the weekend to the total number of communications.
common neighbors	The feature represents the number of common neighbors of v_i and v_j in the mobile network.

REFERENCES

Lada A. Adamic and Eytan Adar. 2001. Friends and neighbors on the web. *Social Networks* 25 (2001), 211–230.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* 6 (2005), 1817–1853.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2006. Multi-task feature learning. In *Proceedings of the 18th Neural Information Processing Systems (NIPS'06)*. 41–48.

Andreas Argyriou, Andreas Maurer, and Massimiliano Pontil. 2008. An algorithm for transfer learning in a heterogeneous environment. In *Proceedings of 2008 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'08)*. 71–85.

Lars Backstrom and Jure Leskovec. 2011. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the 4th ACM International Conference on Web Search and Web Data Mining (WSDM'11)*. 635–644.

Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*. 120–128.

Ronald S. Burt. 1992. *Structural Holes: The Social Structure of Competition*. Harvard University Press.

Bin Cao, Nathan Nan Liu, and Qiang Yang. 2010. Transfer learning for collective link prediction in multiple heterogenous domains. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*. 159–166.

Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. 199–207.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20 (1995), 273–297.

David J. Crandall, Lars Backstrom, Dan Cosley, Daniel Huttenlocher Siddharth Suri, and Jon Kleinberg. 2010. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Science* 107 (Dec. 2010), 22436–22441.

Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2008. Translated learning: Transfer learning across different feature spaces. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS'08)*. 353–360.

Wenyuan Dai, Guirong Xue, Qiang Yang, and Yong Yu. 2007a. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*. 210–219.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007b. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*. 193–200.

James A. Davis and Samuel Leinhardt. 1972. The structure of positive interpersonal relations in small groups. In *Sociological Theories in Progress*, J. Berger (Ed.). Vol. 2. Houghton Mifflin, 218–251.

Christopher P. Diehl, Galileo Namata, and Lise Getoor. 2007. Relationship identification for social network discovery. In *Proceedings of the 24th National Conference on Artificial Intelligence and 9th Conference on Innovative Applications of Artificial Intelligence (AAAI'07)*. 546–552.

Yuxiao Dong, Jie Tang, Nitesh V. Chawla, Tiancheng Lou, Yang Yang, and Bai Wang. 2015. Inferring social status and rich club effects in enterprise communication networks. *PLOS One* 10, 3 (2015), e0119446.

Yuxiao Dong, Jie Tang, Sen Wu, Jilei Tian, Nitesh V. Chawla, Jinghai Rao, and Huanhuan Cao. 2012. Link prediction and recommendation across heterogeneous social networks. In *Proceedings of the 12th International Conference on Data Mining (ICDM'12)*. 181–190.

Yuxiao Dong, Jing Zhang, Jie Tang, Nitesh V. Chawla, and Bai Wang. 2015. CoupledLP: Link prediction in coupled networks. In *Proceedings of the 21nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. 199–208.

Nathan Eagle, Alex (Sandy) Pentland, and David Lazer. 2009. Inferring social network structure using mobile phone data. *Proceedings of the National Academy of Science* 106, 36 (2009), 15274–15278.

David Easley and Jon Kleinberg. 2010. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press.

Jing Gao, Wei Fan, Jing Jian, and Jiawei Han. 2008. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. 283–291.

Lise Getoor and Ben Taskar. 2007. *Introduction to Statistical Relational Learning*. MIT Press.

Malcolm Gladwell. 2001. The Tipping Point - How Little Things Make a Big Difference (New York, NY: Little Brown, 2000); G. Khermouch and J. Green, Buzz Marketing: Suddenly This Stealth Strategy Is Hot – but It's Still Fraught with Risk. *Business Week* (2001), 50.

Bruno Goncalves, Nicola Perra, and Alessandro Vespignani. 2011. Modeling users' activity on twitter networks: Validation of dunbar's number. *PLoS ONE* 6 (08 2011), e22656.

Mark Granovetter. 1973. The strength of weak ties. *American Journal of Sociology* 78, 6 (1973), 1360–1380.

R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web (WWW'04)*. 403–412.

J. M. Hammersley and P. Clifford. 1971. Markov field on finite graphs and lattices. Unpublished manuscript (1971).

John Hopcroft, Tiancheng Lou, and Jie Tang. 2011. Who will follow you back? Reciprocal relationship prediction. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*. 1137–1146.

Tony Jebara. 2004. Multi-task feature and kernel selection for svms. In *Proceedings of the 21th International Conference on Machine Learning (ICML'04)*. 55–62.

George Karypis and Vipin Kumar. 1998. *MeTis: Unstrctured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*. http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/manual.pdf.

Elihu Katz. 1957. The two-step flow of communication: An up-to-date report on an hypothesis. *Public Opinion Quarterly* 21, 1 (1957), 61–78.

Elihu Katz and Paul Felix Lazarsfeld. 1955. *Personal Influence*. The Free Press, New York, USA.

Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 1 (1953), 39–43.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*. 137–146.

Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. 2011. MRF energy minimization and beyond via dual decomposition. *IEEE Transactions on Pattern Analalysis and Machine Intelligence* 33, 3 (2011), 531–552.

David Krackhardt. 1992. The strength of strong ties: The importance of philos in organizations. *Networks and Organizations: Structure, Form, and Action* 216 (1992), 239.

Frank R. Kschischang, Brendan J. Frey, and Hans andrea Loeliger. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* 47 (2001), 498–519.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*. 282–289.

Steffen L. Lauritzen. 1996. *Graphical Models*. Oxford University Press, Oxford.

Paul Felix Lazarsfeld, Bernard Berelson, and Hazel Gaudet. 1944. *The People's Choice: How the Voter Makes up his Mind in a Presidential Campaign*. Columbia University Press, New York, NY, USA.

Su-In Lee, Vassil Chatalbashev, David Vickrey, and Daphne Koller. 2007. Learning a meta-level prior for feature relevance from multiple related tasks. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*. 489–496.

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010a. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. 641–650.

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010b. Signed networks in social media. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI'10)*. 1361–1370.

Xuejun Liao, Ya Xue, and Lawrence Carin. 2005. Logistic regression with an auxiliary data source. In *Proceedings of the 22th International Conference on Machine Learning (ICML'05)*. 505–512.

David Liben-Nowell and Jon M. Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology* 58, 7 (2007), 1019–1031.

Ryan Lichtenwalter, Jake T. Lussier, and Nitesh V. Chawla. 2010. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. 243–252.

Xiao Ling, Guirong Xue, Wenyuan Dai, Yun Jiang, Qiang Yang, and Yong Yu. 2008. Can Chinese web pages be classified with english data source? In *Proceeding of the 17th International Conference on World Wide Web (WWW'08)*. 969–978.

Tiancheng Lou and Jie Tang. 2013. Mining structural hole spanners through information diffusion in social networks. In *Proceedings of the 22th International Conference on World Wide Web (WWW'13)*. 837–848.

Tiancheng Lou, Jie Tang, John Hopcroft, Zhanpeng Fang, and Xiaowen Ding. 2013. Learning to predict reciprocity and triadic closure in social networks. *ACM Transactions on Knowledge Discovery from Data* 7, 2 (2013), Article No. 5.

Julian McAuley and Jure Leskovec. 2014. Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8, 1 (2014), 4.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI'99)*. 467–475.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14, 1 (1978), 265–294.

M. E. J. Newman. 2001. Clustering and preferential attachment in growing networks. *Physical Reviews E* 64, 2 (2001), 025102.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report SIDL-WP-1999-0120. Stanford University.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (Oct. 2010), 1345 –1359.

Maayan Roth, Assaf Ben-David, David Deutscher, Guy Flysher, Ilan Horn, Ari Leichtberg, Naty Leiser, Yossi Matias, and Ron Merom. 2010. Suggesting friends using the implicit social graph. In *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining (KDD'10)*. 233–242.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*. 1070–1079.

Xiaoxiao Shi, Wei Fan, and Jiangtao Ren. 2008. Actively transfer domain knowledge. In *Proceedings of 2008 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'08)*. 342–357.

Xiaodan Song, Yun Chi, Koji Hino, and Belle L. Tseng. 2007. Identifying opinion leaders in the blogosphere. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM'06)*. 971–974.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. 1397–1405.

Chenhao Tan, Jie Tang, Jimeng Sun, Quan Lin, and Fengjiao Wang. 2010. Social action tracking via noise tolerant time-varying factor graphs. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. 1049–1058.

Jie Tang, Tiancheng Lou, and Jon Kleinberg. 2012. Inferring social ties across heterogeneous networks. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM'12)*. 743–752.

Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. 807–816.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. 990–998.

Lei Tang and Huan Liu. 2011. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery* 23, 3 (2011), 447–478.

Wenbin Tang, Honglei Zhuang, and Jie Tang. 2011. Learning to Infer Social Ties in Large Networks. In *Proceedings of 2011 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'11)*. 381–397.

Benjamin Taskar, Ming Fai Wong, Pieter Abbeel, and Daphne Koller. 2003. Link prediction in relational data. In *Proceedings of the 15th Neural Information Processing Systems (NIPS'03)*.

Martin J. Wainwright and Michael I. Jordan. 2008. Graphical models, exponential families, and variational inference. *Foundational Trends in Machine Learning* 1, 1–2 (Jan. 2008), 1–305.

Chi Wang, Jiawei Han, Yuntao Jia, Jie Tang, Duo Zhang, Yintao Yu, and Jingyi Guo. 2010. Mining advisor-advisee relationships from research publication networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. 203–212.

Liaoruo Wang, Tiancheng Lou, Jie Tang, and John Hopcroft. 2011. Detecting community kernels in large social networks. In *Proceedings of 2011 IEEE International Conference on Data Mining (ICDM'11)*. 784–793.

Wim Wiegerinck. 2000. Variational approximations between mean field theory and the junction tree algorithm. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI'00)*. 626–633.

Shaomei Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. 2011. Who says what to whom on twitter. In *Proceedings of the 20th International Conference on World Wide Web (WWW'11)*. 705–714.

Rongjing Xiang, Jennifer Neville, and Monica Rogati. 2010. Modeling relationship strength in online social networks. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. 981–990.

Eric P. Xing, Michael I. Jordan, and Stuart Russell. 2003. A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of 19th Conference on Uncertainty in Artificial Intelligence (UAI'03)*. 583–591.

Yang Yang, Yizhou Sun, Jie Tang, Bo Ma, and Juanzi Li. 2015. Entity matching across heterogeneous sources. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. 1395–1404.

Zi Yang, Jingyi Guo, Keke Cai, Jie Tang, Juanzi Li, Li Zhang, and Zhong Su. 2010. Understanding retweeting behaviors in social networks. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM'10)*. 1633–1636.

Jiawei Zhang, Xiangnan Kong, and Philip S. Yu. 2013. Predicting social links for new users across aligned heterogeneous social networks. In *ICDM'13*. 1289–1294.

Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, and Philip Yu. 2015. COSNET: Connecting heterogeneous social networks with local and global consistency. In *KDD'15*. 1485–1494.

Honglei Zhuang, Jie Tang, Wenbin Tang, Tiancheng Lou, Alvin Chin, and Xia Wang. 2012. Actively learning to infer social ties. *Data Mining and Knowledge Discovery* 25, 2 (2012), 270–297.

Received June 2014; revised January 2015; accepted March 2015