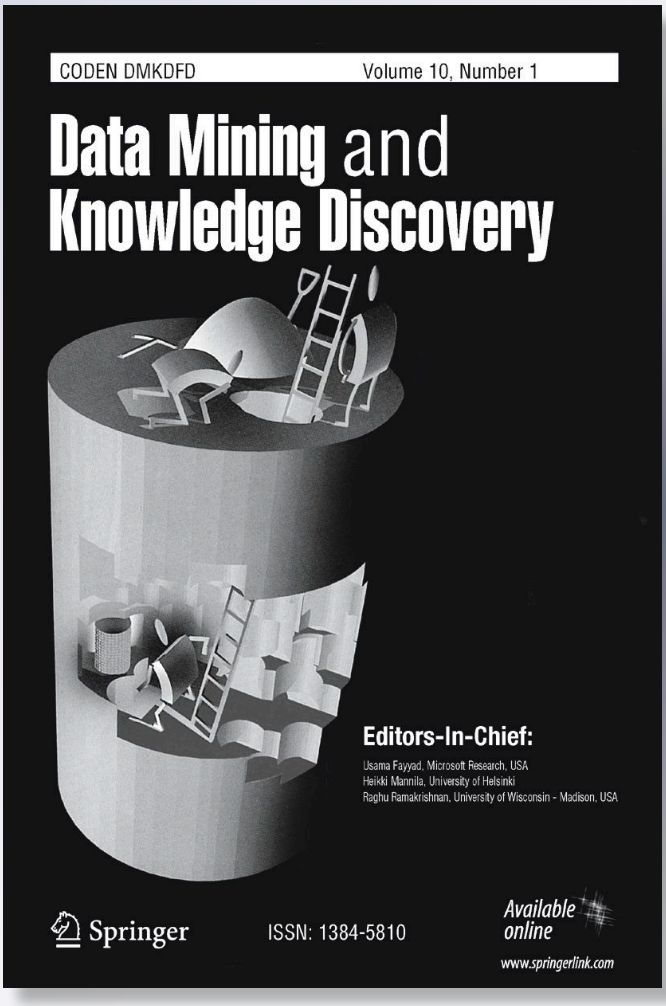
Actively learning to infer social ties

Honglei Zhuang, Jie Tang, Wenbin Tang, Tiancheng Lou, Alvin Chin & Xia Wang

Data Mining and Knowledge Discovery

ISSN 1384-5810
Volume 25
Number 2

Data Min Knowl Disc (2012) 25:270-297
DOI 10.1007/s10618-012-0274-x


CODEN DMKDFD
Volume 10, Number 1
Data Mining and Knowledge Discovery
Editors-In-Chief:
Usama Fayyad, Microsoft Research, USA
Heikki Mannila, University of Helsinki
Raghu Ramakrishnan, University of Wisconsin - Madison, USA
Springer
ISSN: 1384-5810
Available online
www.springerlink.com

Data Min Knowl Disc (2012) 25:270–297
DOI 10.1007/s10618-012-0274-x

Actively learning to infer social ties

Honglei Zhuang · Jie Tang · Wenbin Tang ·
Tiancheng Lou · Alvin Chin · Xia Wang

Received: 31 October 2011 / Accepted: 31 May 2012 / Published online: 22 June 2012

Abstract We study the extent to which social ties between people can be inferred in large social network, in particular via active user interactions. In most online social networks, relationships are lack of meaning labels (e.g., "colleague" and "intimate friends") due to various reasons. Understanding the formation of different types of social relationships can provide us insights into the micro-level dynamics of the social network. In this work, we precisely define the problem of inferring social ties and propose a Partially-Labeled Pairwise Factor Graph Model (PLP-FGM) for learning to infer the type of social relationships. The model formalizes the problem of inferring social ties into a flexible semi-supervised framework. We test the model on three different genres of data sets and demonstrate its effectiveness.We further study how to

Responsible editor: Dimitrios Gunopulos, Donato Malerba and Michalis Vazirgiannis.

H. Zhuang · J. Tang (✉) · W. Tang
Department of Computer Science and Technology, Tsinghua University, Beijing, China
e-mail: jietang@tsinghua.edu.cn

H. Zhuang
e-mail: zhl09@mails.tsinghua.edu.cn

W. Tang
e-mail: tangwb06@gmail.com

T. Lou
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
e-mail: tiancheng.lou@gmail.com

A. Chin · X. Wang
Nokia Research Center, Beijing, China
e-mail: alvin.chin@nokia.com

X. Wang
e-mail: Xia.S.wang@nokia.com

leverage user interactions to help improve the inferring accuracy. Two active learning algorithms are proposed to actively select relationships to query users for their labels. Experimental results show that with only a few user corrections, the accuracy of inferring social ties can be significantly improved. Finally, to scale the model to handle real large networks, a distributed learning algorithm has been developed.

Keywords Social ties · Partially labeled · Factor graph model · Active learning · Influence maximization · Distributed learning

1 Introduction

Online social networks, such as Facebook, MySpace, Twitter, and FourSquare, have already become a bridge to connect our real daily life and the virtual web space. Facebook, one of the largest social networks, has more than 750 million active users in July 2011; Foursquare, a location-based mobile social network, has attracted more than 20 million registered users at the beginning of 2011. Just to mention a few, there is little doubt that most of our friends are online now. Considerable research has been conducted on social network analysis (Albert and Barabasi 2002; Faloutsos 1999; Newman 2003; Strogatz 2003), dynamic evolution analysis (Kleinberg 2005), social influence analysis (Domingos and Richardson 2001; Kempe et al. 2003; Tang et al. 2009), social behavior analysis (Roth et al. 2010; Tan et al. 2010), and social tie analysis (Crandall et al. 2010; Hopcroft et al. 2011; Leskovec et al. 2010; Tang et al. 2012). However, most of these works ignore one important fact that makes the online social networks different from the physical social networks, i.e., our physical social networks are colorful ("family members", "colleagues", and "classmates") but the online social networks are still black-and-white: the users merely do not take the time to label the relationships. Indeed, statistics show that only 16 % of mobile phone users in Europe have created custom contact groups (Roth et al. 2010; Grob et al. 2009) and less than 23 % connections on LinkedIn have been labeled.Understanding the formation of different types of social relationships can provide us insights into the micro-level dynamics of the social network. For example, awareness of the types of social relationships can help many mining applications such as friend recommendation and product advertisement.

In this work, we investigate to what extent social relationships can be inferred from the online social networks. Given users' behavior history and interactions between users, can we estimate how likely they are to be family members? One challenge is how to design a unified model so that it can be easily applied to different domains? There exist a few related studies. For example, Diehl et al. (2007) try to identify the relationships by learning a ranking function. Wang et al. (2010) propose an unsupervised algorithm for mining the advisor-advisee relationships from the publication network. However, Diehl et al. (2007) only considers the communication archive, while Wang et al. (2010) is a domain-specific unsupervised algorithm. Both algorithms are not easy to extend to other domains.

Another challenge is that online social networks are becoming more and more complex and dynamic. Even the best performance achieved by the state-of-the-art algorithms is still under 90 %. The result is unsatisfactory and invariably contains a number of errors. A promising solution is to design an interactive interface to allow users to provide feedbacks on the inferring results. However, we should be aware that the interactive process might be tedious, error-prone, and time-consuming. For example, for inferring advisor-advisee relationships from the coauthor network, an author may have hundreds of coauthors.[1] The user may soon become tired, if she/he is asked to carefully go through all her/his relationships to validate the inferring results. Ideally, an algorithm should be able to actively select only a few potentially wrong relationships to query the user, instead of passively waiting for user feedbacks. The problem is referred to as *actively learning to infer social ties*.

Motivating examples To illustrate the problem, Fig. 1 gives an example of actively inferring ties in a mobile communication network. The left figure gives the input of our problem: a mobile social network, which consists of users, calls made and messages sent between users, and users' attribute information such as location. The objective is to classify the type of social relationships in the network. The middle figure shows the result of the proposed PLP-FGM model, a semi-supervised learning model. The blue solid lines stand for friend relationship between users and the green dash lines indicate colleagues.The probability associated with each relationship represents how confident the learning model is in the inferred type of the relationship. Further, an active learning algorithm selects an uncertain relationship (associated with a question mark) to query the user. Once the user gives the answer, the learning model propagates the correction in the social network and further corrects other relationships (Cf. the right figure).

Therefore, the fundamental problem is how to design a flexible model for effectively and efficiently learning to infer social ties in different networks. This problem is non-trivial and poses a set of unique challenges. First, what are the underlying factors that form a specific type of social relationship. Second, the input social network is partially labeled. We may have some labeled relationships, but most of the relationships are unknown. To learn a high-quality predictive model, we should not only consider the knowledge provided by the labeled relationships, but also leverage the unlabeled network information. Third, how to make optimal use of user interaction. The selection should consider both the uncertainty and the network structure information. Finally, real social networks are getting bigger with thousands even millions of nodes. It is important to develop a method that can scale well to real large networks.

Results In this paper, we try to conduct a systematic investigation for the problem of actively learning to infer social ties in large networks. We precisely define the problem and propose a Partially-Labeled Pairwise Factor Graph Model (PLP-FGM) for solving this problem. To make optimal use of user interactions, two strategies, an influence maximization based strategy and a belief maximization based strategy, have been devised to actively select potentially wrong but most useful relationships to query the user. To scale to large networks, we develop a distributed implementation of the learning algorithm based on MPI (Message-Passing Interface).

[1] An example can be found on http://arnetminer.org/person/jiawei-han-745329.html.

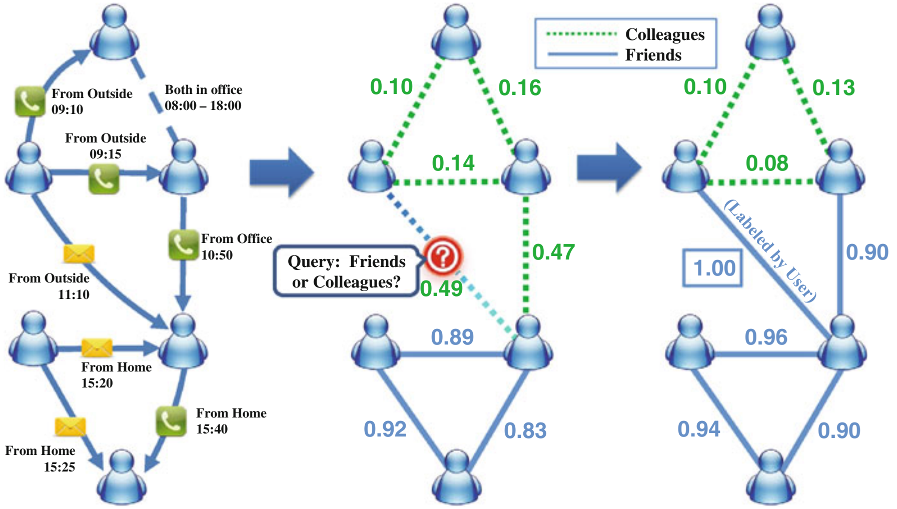


Fig. 1 An example of actively learning to infer social ties in a mobile communication network. The *left* figure is the input of our problem, the *middle* figure shows the inferred relationships by the proposed learning model. The relationship associated with the *question mark* is selected by an active learning algorithm to query the user. The *right* figure is the improved result with the user's feedback

We evaluate the proposed model on three different data sets: Publication, Email and Mobile. Experimental results demonstrate that the proposed PLP-FGM model can accurately infer 92.7 % of advisor-advisee relationships from the coauthor network (Publication), 88.0 % of manager-subordinate relationships from the email network (Email), and 83.1 % of the friendships from the mobile network (Mobile). Our study also reveals several interesting phenomena:

- *Unlabeled data indeed helps*. With the unlabeled relationships, the accuracy of inferring social relationships can be significantly improved (+2.2 to +11.8 %).
- *Strong correlation between relationships.* For example, in the Email network, we obtained 7.6 % improvement on the inferring accuracy by considering the correlations (Co-recipient, Co-manager and Co-subordinate).
- *Network information helps.* For active selection of the relationship, both of the proposed methods which consider the network information outperform (+0.3 to 6.1 %) the alternative baseline methods which consider only the uncertain information.

Organization The rest of paper is organized as follows. Section 2 formally formulates the problem and presents the basic idea of methodologies; Section 3 explains the PLP-FGM model; Section 4 presents the active learning strategies for PLP-FGM; Section 5 gives experimental results; Finally, Section 6 discusses related work and Section 7 concludes.

2 Overview

In this section, we present the problem formulation and the basic ideas of our approach.

2.1 Problem formulation

First, we give several related definitions. A social network can be represented as $G = (V, E)$, where V is a set of $|V| = N$ users and $E \subset V \times V$ is a set of $|E| = M$ relationships between users. The objective of our work is to learn a model that can effectively infer the type of social relationships between two users. More precisely, we first formally define the output of our problem, namely *relationship semantics*.

Definition 1 Relationship semantics: Relationship semantics is a triple (e_{ij}, r_{ij}, p_{ij}), where $e_{ij} \in E$ is a social relationship; $r_{ij} \in \mathcal{Y}$ is a label associated with the relationship; $\mathcal{Y}$ is the set of all the labels; p_{ij} is the probability (confidence) obtained by an algorithm for inferring relationship type.

Social relationships might be undirected in some networks (e.g., the friendship discovered from the mobile communication network) or directed in other networks (e.g., the advisor-advisee relationship in the publication network). To be consistent, we define all social relationships as directed relationships. In addition, relationships may be static (e.g., the family-member relationship) or dynamic over time (e.g., colleague relationship). In this work, we focus on static relationships, and leave the dynamic case to our future work.

To infer relationship semantics, we could consider different factors such as user-specific information, link-specific information, and global constraints (Cf. § 5.1 and § A for examples). For example, to discover advisor-advisee relationships from a publication network, we can consider how many papers were coauthored by two authors; how many papers in total an author has published; when the first paper was published by each author. Besides, there may exist some labeled relationships. Formally, we can define the input of our problem as a partially labeled network.

Definition 2 Partially labeled network: A partially labeled network is an augmented social network denoted as $G = (V, E^L, E^U, R^L, \mathbf{W})$, where E^L is a set of labeled relationships and E^U is a set of unlabeled relationships with $E^L \cup E^U = E$; R^L is a set of labels corresponding to the relationships in E^L; $\mathbf{W}$ is an attribute matrix associated with users in V where each row corresponds to a user, each column an attribute, and an element w_{ij} the value of the j^{th} attribute of user v_i.

Based on the above concepts, we can define the problem of inferring social relationships. Given a partially labeled network, the goal is to detect the types (labels) of all unknown relationships in the network. More precisely,

Problem 1 Social relationship mining. Given a partially labeled network $G = (V, E^L, E^U, R^L, \mathbf{W})$, the objective is to learn a predictive function

$$f : G = (V, E^L, E^U, R^L, \mathbf{W}) \rightarrow R$$

Another important question is how we can learn the mapping function f effectively. In many situations, labeled data is limited and expensive. The problem is, can we design a strategy to *actively* learn the model with minimal labeling cost? Formally,

Problem 2 Active relationship mining. Given a partially labeled network $G = (V, E^L, E^U, R^L, \mathbf{W})$, and a labeling budget b (number of user interactions). Our objective is to select a subset of unknown relationships $A \subset E^U$ within the constraint of b to label, so that the performance of predictive function f can be maximally improved.

Our formulation of inferring social relationships is very different from existing works on relation mining (Califf and Mooney 1999), which focuses on detecting the relationships from the content information, while we focus on mining relationship semantics in social networks. Diehl et al. (2007) and Wang et al. (2010) investigate the problem of relationship identification. However, they study the problem in specific domains (Email network or Publication network). Backstrom and Leskovec (2011) propose an algorithm based on supervised random walks for link prediction. Crandall et al. (2010) incorporate geographic coincidences to infer social ties, while Wang et al. (2011) consider user mobility and network proximity. Different from these works which aim at link prediction, our goal is to infer the types of relationships. There are also works on inferring the types of relationships. Hopcroft et al. (2011) explore the problem of reciprocal relationship prediction and Tang et al. (2012) have developed a framework for classifying the type of social relationships by learning across heterogeneous networks. Yang et al. (2010) study the retweeting behavior. Leskovec et al. (2010) focus on the prediction of edge signs (positive or negative). However, they do not consider how to make optimal use of user interaction.

2.2 Our approach

For inferring the type of social relationships, we have three basic intuitions. First, the user-specific or link-specific attributes will contain implicit information about the relationships. For example, two users who make a number of calls in working hours might be colleagues; while two users who frequently contact with each other in the evening are more likely to be family members or intimate friends. Second, relationships among different users may have a correlation. For example, in the mobile network, if user v_i makes a call to user v_j immediately after calling user v_k, then user v_i may have a similar relationship (family member or colleague) with user v_j and user v_k. Third, we also need to consider some global constraints such as common knowledge or user-specific constraints.

Based on the intuitions above, we propose a Partially-Labeled Pairwise Factor Graph Model (PLP-FGM). It allows us to take all the factors mentioned above into account to better infer the social relationships. We will describe the model in details in Section 3.

For actively selecting helpful relationships to query the user, we define a quality function $Q(A)$, which measures the expected improvement of the prediction performance by labeling relationships in set A. The problem can be then defined as an optimization problem of $Q(A)$, i.e.,

$$A^* = \arg \max_{A \subset Y^U} Q(A), |A| = b, b > 0$$

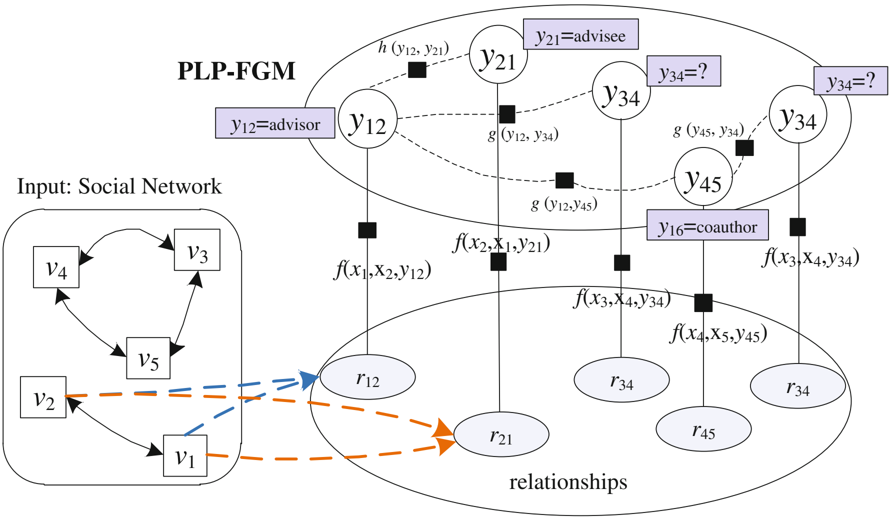


Fig. 2 Graphical representation of the PLP-FGM model

To quantify $Q(A)$, we could consider how a selected node can influence the others. For example, correction of a centered relationship may trigger a spread of the correction, thus help infer correlated relationships.

Based on the above intuitions, we develop an Influence-Maximization Selection (IMS) model and a Belief-Maximization Selection (BMS) model for actively inferring the types of social relationships. The IMS model selects the most influential nodes, by leveraging the network structure and the uncertainty obtained from PLP-FGM. The BMS model further incorporates the active selection process into the learning process of PLP-FGM.

3 Partially-labeled pairwise factor graph model (PLP-FGM)

Typically, there are two ways to model the social tie inferring problem. The first way is to model each user as a node and for each node to estimate the probability distribution of different relationships. The resultant graphical model thus consists of N variable nodes. Each node contains a $d \times |\mathcal{Y}|$ matrix to represent the probability distributions of different relationships between the user and her/his neighbors, where d is the number of neighbors of the node. This model is intuitive, but it suffers from some limitations. For example, it is difficult to model the correlations between two relationships, and its computational complexity is high. An alternative way is to model each relationship as a node in the graphical model and the relationship mining task becomes how to predict the semantic label for each relationship node in the model. This model contains M nodes ($2M$ when the input social network is undirected). This model is able to incorporate different correlations between relationships such as the above intuitions.

We propose a Partially-Labeled Pairwise Factor Graph Model (PLP-FGM). Figure 2 shows the graphical representation of the PLP-FGM. Each relationship (v_{i_1}, v_{i_2}) or $e_{i_1 i_2}$ in the partially labeled network G is mapped to a *relationship node* r_i

in PLP-FGM. We denote the set of relationship nodes as $Y = \{y_1, y_2, \ldots, y_M\}$. The relationships in G are partially labeled, thus all nodes in PLP-FGM can be divided into two subsets Y^L and Y^U, corresponding to the labeled and unlabeled relationships respectively. For each relationship node $y_i = (v_{i_1}, v_{i_2}, r_{i_1 i_2})$, we combine the attributes $\{\mathbf{w}_{i_1}, \mathbf{w}_{i_2}\}$ into a *relationship attribute vector* $\mathbf{x}_i$.

Now we explain the PLP-FGM in details. The relationships in the input are modeled by relationship nodes in PLP-FGM. Corresponding to the three intuitions, we define the following three factors.

- *Attribute factor*: $f(y_i, \mathbf{x}_i)$ represents the posterior probability of the relationship y_i given the attribute vector $\mathbf{x}_i$;
- *Correlation factor*: $g(y_i, G(y_i))$ denotes the correlation between the relationships, where $G(y_i)$ is the set of correlated relationships to y_i.
- *Constraint factor*: $h(y_i, H(y_i))$ reflects the constraints between relationships, where $H(y_i)$ is the set of relationships constrained on y_i.

Given a partially labeled network $G = (V, E^L, E^U, R^L, \mathbf{W})$, we can define the joint distribution over Y as

$$p(Y|G) = \prod_i f(y_i, \mathbf{x}_i) g(y_i, G(y_i)) h(y_i, H(y_i)) \tag{1}$$

The three factors can be instantiated in different ways. In this paper, we use exponential-linear functions. In particular, we define the attribute factor as

$$f(y_i, \mathbf{x}_i) = \frac{1}{Z_\lambda} \exp\{\lambda^T \Phi(y_i, \mathbf{x}_i)\} \tag{2}$$

where λ is a weighting vector and Φ is a vector of feature functions. Similarly, we define the correlation factor and constraint factor as

$$g(y_i, G(y_i)) = \frac{1}{Z_\alpha} \exp\{\sum_{y_j \in G(y_i)} \alpha^T \mathbf{g}(y_i, y_j)\} \tag{3}$$

$$h(y_i, H(y_i)) = \frac{1}{Z_\beta} \exp\{\sum_{y_j \in H(y_i)} \beta^T \mathbf{h}(y_i, y_j)\} \tag{4}$$

where $\mathbf{g}$ and $\mathbf{h}$ can be defined as a vector of indicator functions. This feature definition was often used in a graphical models such as Markov Random Fields (Hammersley and Clifford 1971) or Conditional Random Fields (Lafferty et al. 2001).

Model learning Learning PLP-FGM is to estimate a parameter configuration $\theta = (\lambda, \alpha, \beta)$, so that the log-likelihood of observation information (labeled relationships) are maximized. For presentation simplicity, we concatenate all factor functions for a relationship node y_i as $\mathbf{s}(y_i) = (\Phi(y_i, \mathbf{x}_i)^T, \sum_{y_j} \mathbf{g}(y_i, y_j)^T, \sum_{y_j} \mathbf{h}(y_i, y_j)^T)^T$. The joint probability defined in (Eq. 1) can be rewritten as

$$p(Y|G) = \frac{1}{Z} \prod_i \exp\{\theta^T \mathbf{s}(y_i)\} = \frac{1}{Z} \exp\{\theta^T \sum_i \mathbf{s}(y_i)\} = \frac{1}{Z} \exp\{\theta^T \mathbf{S}\} \tag{5}$$

```
Input: learning rate $\eta$
Output: learned parameters $\theta$
Initialize $\theta$;
repeat
    Calculate $\mathbb{E}_{p_\theta(Y|Y^L,G)}\mathbf{S}$ using LBP ;
    Calculate $\mathbb{E}_{p_\theta(Y|G)}\mathbf{S}$ using LBP ;
    Calculate the gradient of $\theta$ according to Eq. 7:
        $\nabla_\theta = \mathbb{E}_{p_\theta(Y|Y^L,G)}\mathbf{S} - \mathbb{E}_{p_\theta(Y|G)}\mathbf{S}$
    Update parameter $\theta$ with the learning rate $\eta$:
        $\theta_{\text{new}} = \theta_{\text{old}} - \eta \cdot \nabla_\theta$
until Convergence;
```

Algorithm 1: Learning PLP-FGM

where $Z = Z_\lambda Z_\alpha Z_\beta$ is a normalization factor (also called partition function), $\mathbf{S}$ is the aggregation of factor functions over all relationship nodes, i.e., $\mathbf{S} = \sum_i \mathbf{s}(y_i)$.

One challenge for learning the PLP-FGM model is that the input data is partially labeled. To calculate the partition function Z, one needs to sum up the likelihood of possible states for all nodes including unlabeled nodes. To deal with this, we use the labeled data to infer unknown labels. Here $Y|Y^L$ denotes a labeling configuration Y inferred from the known labels. Thus, we can define the following log-likelihood objective function $\mathcal{O}(\theta)$:

$$\begin{aligned}\mathcal{O}(\theta) &= \log p(Y^L|G) = \log \sum_{Y|Y^L} \frac{1}{Z} \exp\{\theta^T \mathbf{S}\} \\ &= \log \sum_{Y|Y^L} \exp\{\theta^T \mathbf{S}\} - \log Z \\ &= \log \sum_{Y|Y^L} \exp\{\theta^T \mathbf{S}\} - \log \sum_{Y} \exp\{\theta^T \mathbf{S}\} \qquad (6)\end{aligned}$$

To solve the objective function, we consider a gradient decent method (or a Newton–Raphson method). Specifically, we first calculate the gradient for each parameter θ:

$$\begin{aligned}\frac{\partial \mathcal{O}(\theta)}{\partial \theta} &= \frac{\partial \left(\log \sum_{Y|Y^L} \exp \theta^T \mathbf{S} - \log \sum_{Y} \exp \theta^T \mathbf{S}\right)}{\partial \theta} \\ &= \frac{\sum_{Y|Y^L} \exp \theta^T \mathbf{S} \cdot \mathbf{S}}{\sum_{Y|Y^L} \exp \theta^T \mathbf{S}} - \frac{\sum_{Y} \exp \theta^T \mathbf{S} \cdot \mathbf{S}}{\sum_{Y} \exp \theta^T \mathbf{S}} \\ &= \mathbb{E}_{p_\theta(Y|Y^L,G)}\mathbf{S} - \mathbb{E}_{p_\theta(Y|G)}\mathbf{S} \qquad (7)\end{aligned}$$

Another challenge here is that the graphical structure in PLP-FGM can be arbitrary and may contain cycles, which makes it intractable to directly calculate the expectation $\mathbb{E}_{p_\theta(Y|G)}\mathbf{S}$. A number of approximate algorithms have been proposed, such as Loopy Belief Propagation (LBP) (Murphy et al. 1999). In this paper, we utilize Loopy Belief Propagation. Specifically, we approximate marginal probabilities $p(y_i|\theta)$ and $p(y_i, y_j|\theta)$ using LBP. With the marginal probabilities, the gradient can be obtained by summing over all relationship nodes. It is worth noting that we need to perform the LBP process twice in each iteration, one time for estimating the marginal probability $p(y|G)$ and the other for $p(y|Y^L, G)$. Finally with the gradient, we update each parameter with a learning rate η. The learning algorithm is summarized in Algorithm 1.

Inferring unknown social ties We now turn to describe how to infer the type of unknown social relationships. Based on learned parameters θ, we can predict the label of each relationship by finding a label configuration which maximizes the joint probability (Eq. 1), i.e.,

$$Y^* = \text{argmax}_{Y|Y^L} p(Y|G) \tag{8}$$

Again, we utilize the Loopy Belief Propagation (LBP) to compute the marginal probability of each relationship node $p(y_i|Y^L, G)$ and then predict the type of a relationship as the label with the largest marginal probability. The marginal probability is taken as the prediction confidence.

Time complexity analysis We use ν_1, ν_2, ν_3 to denote the number of attribute factors, correlation factors and constraint factors in our PLP-FGM respectively. In each round of LBP, the time cost of propagation is $\mathcal{O}(\nu_1 \cdot \dim(\Phi) + \nu_2 \cdot \dim(\mathbf{g}) + \nu_3 \cdot \dim(\mathbf{h}))$, where $\dim(\cdot)$ is the dimension of a vector. We execute the learning algorithm for n iterations, and in each round we execute LBP for n_{LBP} iterations. Thus we can estimate the time complexity as $\mathcal{O}((\nu_1 \cdot \dim(\Phi) + \nu_2 \cdot \dim(\mathbf{g}) + \nu_3 \cdot \dim(\mathbf{h})) \times n \times n_{LBP})$.

4 Actively learning PLP-FGM

The quality function $Q(A)$ can be defined in different forms. Without any constraints, optimizing the quality function $Q(A)$ needs to enumerate all possible subsets $A \subset Y^U$, which is obviously NP-hard. Let us first review two baseline greedy algorithms; and then we will present two new algorithms: Influence-Maximization Selection model (IMS) and Belief-Maximization Selection model (BMS); a theoretical analysis will be given.

4.1 Baseline methods

Maximum uncertainty (MU) A most common selection strategy for active learning is to select the most uncertain relationships. The uncertainty of an unlabeled relationship y_i is measured by the *entropy* $H(y_i) = -\sum_{y \in \mathcal{Y}} p(y_i = y) \log p(y_i = y)$. Based on this intuition, we can define the quality function as

$$Q_{MU}(A) = H(A) \tag{9}$$

where $H(A) = \sum_{y_i \in A} H(y_i)$.

Information density (ID) A drawback of the Maximum Uncertainty strategy is its tendency to choose outliers. Thus we employ another strategy, Information Density, proposed in Settles and Craven (2008). The idea is to choose the most *representative* nodes in Y^U, which are supposed to be the most informative ones. Based on this intuition, we measure the informativeness of a node by its cosine similarity to all other unlabeled nodes in the sense of the attributes attached to a node. Formally, we define the quality function as

$$Q_{ID}(A) = \sum_{i \in A} H(y_i) \times [\frac{1}{|Y^U|} \sum_{j \in Y^U} \text{sim}(\mathbf{x}_i, \mathbf{x}_j)] \tag{10}$$

where $\text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \times \|\mathbf{x}_j\|}$. Note that we again employ the entropy of a relationship node $H(y_i)$ to leverage the "base" informativeness.

4.2 Proposed methods

Influence-maximization selection (IMS) All the strategies mentioned above do not consider the network structure information. As relationship nodes in PLP-FGM are correlated, the most influential nodes are more likely to help improve the overall performance of the model. Existing work has studied several influence propagation model, including the Linear Threshold Model (LTM) in Kempe et al. (2003). The LTM model sets a threshold value ε_i for each node, and weights $b_{i,j}$ for its edges, satisfying $\sum_{j \in NB(i)} b_{i,j} \leq 1$. In each time stamp, 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 node reflecting the strength of the influence spreading in our model. The propagation process is described as:

- The graph is the same as the PLP-FGM model. In addition, we call a relationship node as "activated" when its label y_i is determined. The initial activated set of nodes is 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 node. Thus a node with higher uncertainty will be easier to be activated.
- When a node i is activated, it spreads its gained score increment $(g_i - \varepsilon_i)$ to its neighbor nodes $j \in NB(i)$ with a weight $b_{i,j}$, i.e. $g_j \leftarrow g_j + b_{i,j}(g_i - \varepsilon_i)$. The gained score increment reflects the improvement of confidence brought by user labeling, therefore the influence by labeling an uncertain relationship will be greater than labeling a more certain relationship. To simplify the problem, we set weight $b_{i,j} = 1/|NB(j)|$.
- If a node is labeled by the user, we set it as activated and assign its gained score as 1. The gained score for other nodes is set to 0 at the beginning. Once an inactivated node k gains a score which exceeds the threshold, i.e. $g_k > \varepsilon_i$, it will become activated and spread its gained score similarly. An activated node only spreads its gained score once and remains its status.

We define the quality function $Q_{IMS}(A)$ as the total number of activated nodes after the propagation process. Finding the set A that maximizes the quality function

$Q_{IMS}(A)$ is NP-hard. Similar to Kempe et al. (2003), in this paper, we use a greedy strategy to approximate the solution. Note that unlike the LTM, we do not guarantee a lower bound of error for the greedy optimization method.

Belief-maximization selection (BMS) To quantify the influence of one node on the others, we employ the *belief* of each node obtained by Loopy Belief Propagation in our model. We define a heuristic by removing the effect of attributes from the belief score, denoted by $\mathcal{B}(y_i|G, Y^L)$. More precisely,

$$\mathcal{B}(y_i|G, Y^L) = \exp\{\theta^T \mathbf{s}(y_i) - \lambda^T \Phi(y_i, \mathbf{x}_i)\}$$

By normalizing the belief of one relationship node, we obtain the *belief marginal probability*.

$$p_{\mathcal{B}}(y_i|G, Y^L) = \frac{1}{Z_{\mathcal{B}}}\mathcal{B}(y_i|G, Y^L)$$

where $Z_{\mathcal{B}}$ is the normalization factor. It estimates the marginal probability distribution of a relationship node where the information of its attribute vector is absent.

A basic intuition is, the belief of a relationship node is *monotonically increasing* with respect to the number of relationship nodes of the same type, i.e., $\mathcal{B}(y_i = y|G, Y^L)$ is monotonically increasing with respect to the number of relationships with label y.[2] Without loss of generality, we first consider the binary relationship mining problem, i.e., there are only two possible labels of relationships ($\mathcal{Y} = \{0, 1\}$). In the binary setting, we further consider the active selection for each type separately. This is because when mixing the different types of relationships together, it cannot be guaranteed to have a closed-form solution. Thus, when users provide only positive feedback, our objective is to find a set of positive nodes. Accordingly, we define the quality function of the *positive-oriented BMS strategy* as:

$$Q_{BMS^+}(A) = \sum_{y_i \in Y^U_{(1)}} p_{\mathcal{B}}(y_i = 1|G, Y^L \cup A) \tag{11}$$

where $Y^U_{(1)} = \{y_i | y_i \in Y^U \wedge \mathcal{B}(y_i = 1|G, Y^L) \geq \mathcal{B}(y_i = 0|G, Y^L)\}$.

Symmetrically, if the users provide only *negative* feedback, we can adopt a *negative-oriented BMS strategy*, with the following quality function:

$$Q_{BMS^-}(A) = \sum_{y_i \in Y^U_{(0)}} p_{\mathcal{B}}(y_i = 0|G, Y^L \cup A) \tag{12}$$

The optimization of both quality functions $Q_{BMS^+}(A)$ and $Q_{BMS^-}(A)$ is NP-hard. However, as both quality functions are submodular (theoretical analysis is given in

[2] We present a sufficient condition for this assumption. If for all $y' \in \mathcal{Y}, y' \neq y$, we can have $\exp\{\alpha^T \mathbf{g}(y, y) + \beta^T \mathbf{h}(y, y)\} \geq \exp\{\alpha^T \mathbf{g}(y, y') + \beta^T \mathbf{h}(y, y')\}$, then $\mathcal{B}(y_i = y|G, Y^L)$ is monotonically increasing with respect to the number of y-labeled relationships in Y^L.

```
Input: $G$, $b$
Output: a set of selected relationships $A$
Train PLP-FGM and get the parameter configuration $\theta$;
$A^+, A^- \leftarrow \emptyset$;
for $b/2$ times do
    Use Loopy Belief Propagation(LBP) to obtain the probability distribution for each relationship;
    Find $y_{max^+} = \text{argmax}_{y_i \in Y^U} p(y_i = 1|G, Y^L)(Q_{BMS^+}(A^+ \cup y_i) - Q_{BMS^+}(A^+))$;
    Move $y_{max^+}$ from $Y^U$ to $A^+$;
    $Y^L \leftarrow Y^L \cup A^+$;
    Use Loopy Belief Propagation(LBP) to obtain the probability distribution for each relationship;
    Find $y_{max^-} = \text{argmax}_{y_i \in Y^U} p(y_i = 0|G, Y^L)(Q_{BMS^-}(A^- \cup y_i) - Q_{BMS^-}(A^-))$;
    Move $y_{max^-}$ from $Y^U$ to $A^-$;
    $Y^L \leftarrow Y^L \cup A^-$;
end
```

Algorithm 2: Belief-Maximization selection

§4.3), a solution with an approximation ratio of $(1 - 1/e)$ can be obtained using a greedy algorithm: at each time, it selects the relationship which is expected to provide the maximum marginal increase of the quality function. Notice that we treat the examining relationship node y_i as if it is positive-labeled when optimizing $Q_{BMS^+}(A)$, or negative-labeled for $Q_{BMS^-}(A)$, since the active learning algorithm is label-unaware in the selection stage. In order to leverage the risk that a selected relationship is not labeled as expected, we employ a weighting factor $p(y_i|G, Y^L)$ to reflect how likely the relationship would be labeled as positive(negative).

To prevent making an imbalance selection, we intuitively use Q_{BMS^+} to choose $b/2$ nodes (where b is the number of relationships we expect to query the user each time), and then use Q_{BMS^-} for the rest. Algorithm 2 formally describes the selection process. This selection strategy is denoted by BMS. Note that BMS combines both BMS+ and BMS−. Thus it cannot guarantee a lower error bound for the approximation.

4.3 Theoretical analysis

We give a theoretical analysis of proposed active learning models. The approximation ratio of the IMS model is given in Kempe et al. (2003). Here we focus on the proof of approximation guarantees of the BMS model. The proof is based on the submodular property, which indicates that the marginal gain from adding an element to a set S is at least as high as the marginal gain from adding the same element to a superset of S. The following is a formal definition of the submodular set function.

Definition 3 (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 $F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B)$.

Given a submodular function F, which is also monotone and non-negative, it is an NP-hard problem to find a k-element subset S to optimize F. But a greedy algorithm can result in an approximation ratio of $(1 - 1/e)$. It constructs the subset by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Thus, we have,

Theorem 1 *For a non-negative, monotone submodular function F, let S be the k-element subset decided by the following algorithm: for k times, each time choose an element which gives the maximum marginal increase of F and move it to S. Let S^* denotes the optimal solution. Then we have $F(S) \geq (1 - \frac{1}{e})F(S^*)$.*

Before we prove the submodularity of the quality function Q_{BMS^+}, we first prove the monotonicity of function $p_{\mathcal{B}}(y_i = 1|S)$.

Lemma 1 *For all $y_i \in Y^U$, function $p_{\mathcal{B}}(y_i = 1|S)$ is monotonic with respect to S.*

Suppose x is another unlabeled relationship. We have

$$\begin{aligned} p_{\mathcal{B}}(y_i = 1|S \cup \{x\}) &= \frac{\mathcal{B}(y_i = 1|S \cup \{x\})}{\mathcal{B}(y_i = 1|S \cup \{x\}) + \mathcal{B}(y_i = 0|S \cup \{x\})} \\ &= \frac{1}{1 + \frac{\mathcal{B}(y_i=0|S\cup\{x\})}{\mathcal{B}(y_i=1|S\cup\{x\})}} \end{aligned}$$

Let $k_1 = \frac{\mathcal{B}(y_i=0|S\cup\{x\})}{\mathcal{B}(y_i=1|S\cup\{x\})}$, then

$$p_{\mathcal{B}}(y_i = 1|S \cup \{x\}) = \frac{1}{1 + k_1}$$

Similarly, let $k_2 = \frac{\mathcal{B}(y_i=0|S)}{\mathcal{B}(y_i=1|S)}$, then

$$p_{\mathcal{B}}(y_i = 1|S) = \frac{1}{1 + k_2}$$

According to the assumption in 4.2, we have $k_1 \leq k_2$. Obviously,

$$p_{\mathcal{B}}(y_i = 1|S \cup \{x\}) \geq p_{\mathcal{B}}(y_i = 1|S)$$

Now we prove the submodularity of the quality function Q_{BMS^+} defined by Eq. 11.

Theorem 2 *The quality function $Q_{BMS^+}(S)$ satisfies the submodular property, when $S \subset Y^U_{(1)}$.*

Proof The first step is to prove that function $F(S) = p(y_i = 1|G, Y^L \cup S)$ is submodular with respect to S. Suppose $A \subset B \subset Y^U_{(1)}$, and there is another unlabeled relationship $x \notin B$.

Similarly, we define k_1, k_2, k_3, k_4 below:

$$\begin{aligned} k_1 &= \frac{\mathcal{B}(y_i = 0|G, Y^L \cup A \cup \{x\})}{\mathcal{B}(y_i = 1|G, Y^L \cup A \cup \{x\})}, \quad & k_2 &= \frac{\mathcal{B}(y_i = 0|G, Y^L \cup A)}{\mathcal{B}(y_i = 1|G, Y^L \cup A)} \\ k_3 &= \frac{\mathcal{B}(y_i = 0|G, Y^L \cup B \cup \{x\})}{\mathcal{B}(y_i = 1|G, Y^L \cup B \cup \{x\})}, \quad & k_4 &= \frac{\mathcal{B}(y_i = 0|G, Y^L \cup B)}{\mathcal{B}(y_i = 1|G, Y^L \cup B)} \end{aligned}$$

Since $A \subset B \subset Y^U_{(1)}$, we have $k_1, k_2, k_3, k_4 \leq 1$. In addition,[3] since our factor functions are defined as exponential-linear functions, we can have $k_1/k_2 = k_3/k_4$. We define α and β as follows,

$$\alpha = \frac{k_1}{k_2} = \frac{k_3}{k_4} \leq 1, \beta = \frac{k_3}{k_1} = \frac{k_4}{k_2} \leq 1$$

Then we can obtain the following inequality,

$$\begin{aligned}
\delta(A, x) &= p_{\mathcal{B}}(y_i = 1|G, Y^L \cup A \cup \{x\}) - p_{\mathcal{B}}(y_i = 1|G, Y^L \cup A) \\
&= \frac{1}{1 + k_1} - \frac{1}{1 + k_2} \\
&= \frac{(1 - \alpha)k_2}{(1 + \alpha k_2)(1 + k_2)} \\
\delta(B, x) &= p_{\mathcal{B}}(y_i = 1|G, Y^L \cup B \cup \{x\}) - p_{\mathcal{B}}(y_i = 1|G, Y^L \cup B) \\
&= \frac{(1 - \alpha)k_4}{(1 + \alpha k_4)(1 + k_4)} \\
&= \delta(A, x)\frac{(1 + \alpha)\beta k_2 + \beta + \alpha\beta k_2^2}{(1 + \alpha)\beta k_2 + 1 + \alpha\beta^2 k_2^2} \\
&\leq \delta(A, x)
\end{aligned}$$

Then we give the proof of the submodularity of quality function Q_{BMS^+}. Suppose $A \subset B \subset Y^U$, and there is another unlabeled relationship $y_x \notin B$.

$$\begin{aligned}
\Delta(A, x) &= Q_{BMS^+}(A \cup \{y_x\}) - Q_{BMS^+}(A) \\
&= \sum_{y_i \in Y^U_{(1)}} p_{\mathcal{B}}(y_i = 1|G, Y^L \cup A \cup \{y_x\}) - \sum_{y_i \in Y^U_{(1)}} p_{\mathcal{B}}(y_i = 1|G, Y^L \cup A) \\
&= \sum_{y_i \in Y^U_{(1)} \backslash (A \cup \{y_x\})} [\delta(A, y_x)] + 1 - p_{\mathcal{B}}(y_x = 1|G, Y^L \cup A) \\
&\geq \sum_{y_i \in Y^U_{(1)} \backslash (B \cup \{y_x\})} [\delta(B, y_x)] + 1 - p_{\mathcal{B}}(y_x = 1|G, Y^L \cup B) \\
&= \Delta(B, x)
\end{aligned}$$

□

Therefore, we have proved that Q_{BMS^+} is submodular. The submodularity of Q_{BMS^-} can be proved in a similar way. According to Theorem 1, it guarantees a lower bound of the greedy algorithm employed for the BMS model.

[3] If there is no factor functions between x and y_i, the conclusion $k_1/k_2 = k_3/k_4$ is obvious; otherwise, $\mathcal{B}(y_i|G, Y^L \cup S \cup \{x\})/\mathcal{B}(y_i|G, Y^L \cup S)$ is only relevant to the factor function between x and y_i, and the conclusion can be derived accordingly.

Table 1 Statistics of three data sets

Data set	Users	Unlabeled relationships	Labeled relationships
Publication	1,036,990	1,984,164	6,096
Email	151	3,424	148
Mobile	107	5,122	314

5 Experimental results

The proposed relationship mining approach is general and can be applied to many different scenarios. In this section, we present experiments on three different genres of data sets to evaluate the effectiveness and efficiency of our proposed approach. All data sets and codes are publicly available.[4]

5.1 Experiment setup

Data sets We evaluate the proposed methods on three different data sets: Publication, Email, and Mobile. Statistics of the data sets are listed in Table 1.

- Publication. In the publication data set, we try to infer the advisor-advisee relationship from the coauthor network. The data set is provided by Wang et al. (2010). Specifically, we have collected 1,632,442 publications from Arnetminer (Tang et al. 2008) (from 1936 to 2010) with 1,036,990 authors involved. The ground truth is obtained in three ways: (1) manually crawled from researcher's homepage; (2) extracted from Mathematics Genealogy project;[5] (3) extracted from AI Genealogy project.[6] In total, we have collected 2,164 advisor-advisee pairs as positive cases, and another 3,932 pairs of colleagues as negative cases. The mining results for advisor-advisee relationships are also available in the online system Arnetminer.org.
- Email. In the email data set, we aim to infer the manager-subordinate relationship from the email communication network. The data set consists of 136,329 emails between 151 Enron employees. The ground truth of manager-subordinate relationships is provided by Diehl et al. (2007).
- Mobile. In the mobile data set, we try to infer the friendship in mobile calling network. The data set is from Eagle et al. (2009). It consists of call logs, bluetooth scanning logs and location logs collected by a software installed in mobile phones of 107 users during a ten-month period. In the data set, users provide labels for their friendships. In total, 314 pairs of users are labeled as friends.

Factor definition In the Publication data set, relationships are established between authors v_i and v_j if they coauthored at least one paper. For each pair of coauthors

[4] http://arnetminer.org/socialtie/.

[5] http://www.genealogy.math.ndsu.nodak.edu.

[6] http://aigp.eecs.umich.edu.

(v_i, v_j), our goal is to identify whether v_i is the advisor of author v_j. In this data set, we consider two types of correlations: 1) *co-advisee*. The assumption is based on the fact that one could have only a limited number of advisors in her/his research career. Based on this, we define a correlation factor h_1 between nodes r_{ij} and r_{kj}. 2) *co-advisor*. Another observation is that if v_i is the advisor of v_j (i.e., $r_{ij} = 1$), then v_i is very likely to be the advisor of some other student v_k who is similar to v_j. We define another factor function h_2 between nodes r_{ij} and r_{ik}.

In the Email data set, we try to discover the "manager-subordinate" relationship. A relationship (v_i, v_j) is established when two employees have at least one email communication. There are in total 3,572 relationships among which 148 are labeled as manager-subordinate relationships. We try to identify the relationship types from the email traffic network. For example, if most of an employee's emails were sent to the same person, then the recipient is very likely to be her manager. A correlation named *co-recipient* is defined, that is, if a user v_i sent more than ϑ emails of which recipients including both v_j and v_k (ϑ is a threshold and is set as 10 in our experiment), then, the relationship r_{ij} and r_{ik} are very likely to be the same. Therefore, a correlation factor is added between the two relationships. Two constraints named *co-manager* and *co-subordinate* are also introduced in an analogous way as that for the publication data.

In the Mobile data set, we try to identify whether two users have a friendship if there were at least one voice call or one text message sent from one to the other. Two kinds of correlations are considered: (1) *co-location*. If more than three users arrived at the same location roughly the same time, we establish correlations between all the relationships in this groups. (2) *related-call*. When v_i makes a call to both v_k and v_j from the same location, or makes a call to v_k immediately after the call with v_j, we add a related-call correlation factor between r_{ij} and r_{ik}.

In addition, we also consider some other factors in the three data sets. A detailed description of the factor definition for each data set is given in Table 6 in Appendix.

Comparison methods We compare our approach with the following methods for inferring relationship types:

SVM: It uses the relationship attribute vector $\mathbf{x}_i$ to train a classification model, and predict the relationships by employing the classification model. We use the SVM-light package to implement SVM.

TPFG: It is an unsupervised method proposed in Wang et al. (2010) for mining advisor-advisee relationships in publication network. This method is domain-specific and thus we only compare with it on the Publication data set.

PLP-FGM-S: The proposed PLP-FGM is based on the partially-labeled network. An alternative strategy is to train the model (parameters) with the labeled nodes only. We use this method to evaluate the necessity of the partial learning.

Evaluation measures To quantitatively evaluate the proposed method, we consider two aspects: performance and scalability. For the relationship mining performance, we consider two-fold cross-validation (i.e., half training and half testing) and evaluate the approaches in terms of accuracy, precision, recall, and F1-score. For scalability, we examine the execution time of the model learning.

Table 2 Performance of relationship mining with different methods on three data sets: publication, email and mobile (%)

Data set	Method	Accuracy	Precision	Recall	F1-score
Publication	SVM	76.6	72.5	54.9	62.1
	TPFG	81.2	82.8	**89.4**	86.0
	PLP-FGM-S	84.1	77.1	78.4	77.7
	PLP-FGM	**92.7**	**91.4**	87.7	**89.5**
Email	SVM	82.6	79.1	**88.6**	83.6
	PLP-FGM-S	85.6	85.8	85.6	85.7
	PLP-FGM	**88.0**	**88.6**	87.2	**87.9**
Mobile	SVM	80.0	**92.7**	64.9	76.4
	PLP-FGM-S	80.9	88.1	71.3	78.8
	PLP-FGM	**83.1**	89.4	**75.2**	**81.6**

Bold values indicate the best performance

All the codes are implemented in C++, and all experiments are conducted on a server running Windows Server 2008 with Intel Xeon CPU E7520 1.87GHz (16 cores) and 128 GB memory. The distributed learning algorithm is implemented on MPI (Message Passing Interface).

5.2 Accuracy performance

Table 2 lists the accuracy performance of inferring the type of social relationships by the different methods.

Performance comparison Our method consistently outperforms other comparative methods on all the three data sets. In the Publication data set, PLP-FGM achieves a +27 % (in terms of F1-score) improvement compared with SVM, and outperforms TPFG by 3.5 % (F1-score) and 11.5 % in terms of accuracy. We observe that TPFG achieves the best recall among all the four methods. This is because that TPFG tends to predict more positive cases (i.e., inferring more advisor-advisee relationships in the coauthor network), thus hurts the precision. As a result, TPFG underperforms our method by 8.6 % in terms of precision. In Email and Mobile data set, PLP-FGM outperforms SVM by +4 % and +5 % respectively.

Unlabeled data indeed helps From the result, it is clearly shown that by utilizing the unlabeled data, our model indeed obtains a significant improvement. Without using the unlabeled data, our model (PLP-FGM-S) results in a large performance reduction (-11.8 % in terms of F1-score) on the publication data set. On the other two data sets, we also observe a clear performance reduction.

Factor contribution analysis We perform an analysis to evaluate the contribution of different factors defined in our model. We first remove all the correlation/constraint factors and only keep the attribute factor, and then add each of the factors into the model and evaluate the performance improvement by each factor. Table 3 shows the

Table 3 Factor contribution analysis on three data sets (%)

Data set	Factors used	Accuracy	Precision	Recall	F1-score
Publication	Attributes	77.1	71.1	59.8	64.9
	+ Co-advisor	83.5	80.9	69.8	75.0 (+10.1 %)
	+ Co-advisee	83.1	79.7	70.2	74.7 (+9.8 %)
	All	92.7	91.4	87.7	89.5(+24.6 %)
Email	Attributes	80.1	79.5	81.2	80.3
	+ Co-recipient	80.8	81.5	79.7	80.6 (+0.3 %)
	+ Co-manager	83.1	82.8	83.5	83.2 (+2.9 %)
	+ Co-subordinate	85.0	84.4	85.7	85.0 (+4.7 %)
	All	88.0	88.6	87.2	87.9 (+7.6 %)
Mobile	Attributes	81.8	88.6	73.3	80.2
	+ Co-location	82.2	89.2	73.3	80.4 (+0.2 %)
	+ Related-call	81.8	88.6	73.3	80.2 (+0.0 %)
	All	83.1	89.4	75.2	81.6 (+1.4 %)

result of factor analysis. We see that almost all the factors are useful for inferring the social relationships, but the contribution is very different. For example, for inferring the manager-subordinate relationship, the co-subordinate factor is the most useful factor which achieves a 4.7 % improvement by F1-score, and the co-manager factor achieves a 2.9 % improvement, while the co-recipient factor only results in a 0.3 % improvement. By combining all the factors together, we can further obtain a 2.9 % improvement. An extreme phenomenon appears on the Mobile data set. With each of the two factors (co-location and related-call), we cannot obtain a clear improvement (0.2 and 0.0 % by F1). However, when combining the two factors and the attribute factor together, we can achieve a 1.4 % improvement, seven times higher than that obtained by the separated case. This is because our model not only considers different factors, but also considers the correlation between them.

5.3 Active learning performance

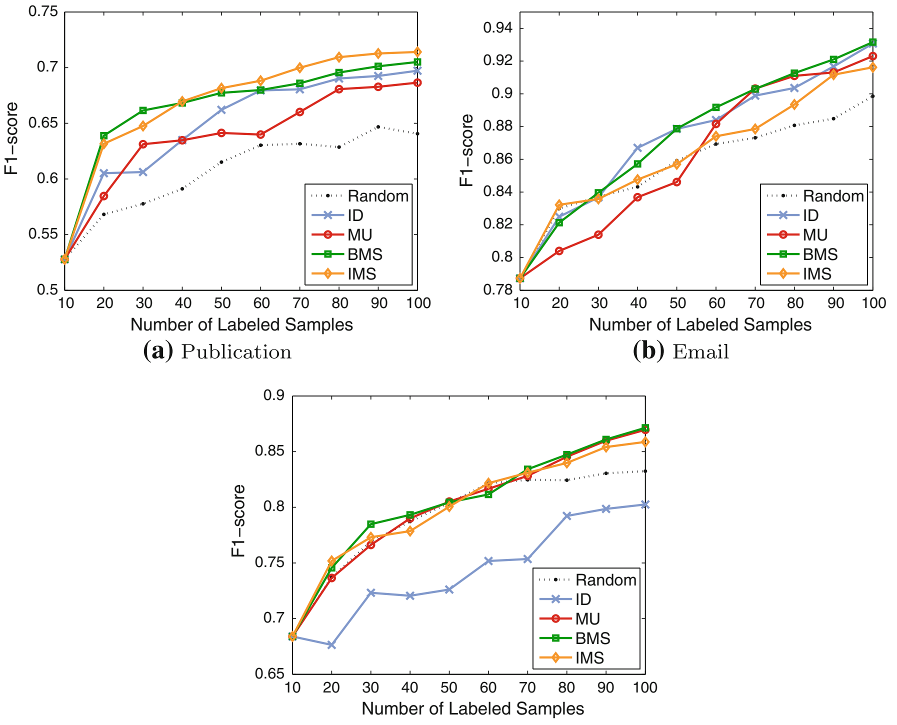
For active learning, in each data set, we first randomly select 10 relationships as the initial labeled set Y^L. And then we iteratively perform the active selection algorithm, each time selecting $b = 10$ relationships to query. After each round of selection, we learn the PLP-FGM model and evaluate the prediction performance. We implement the experiment for 10 times on each data set and use the mean of F1-score for evaluation.

Comparison methods We consider the following baseline methods[7]:

Random: It randomly selects b nodes in Y^U at each time.

Maximum Uncertainty (MU): It chooses the most b uncertain nodes among unlabeled relationships Y^U.

[7] We did not consider the *co-advisee* correlation in the model when dealing with the Publication data set and the *co-subordinate* correlation for the Email data set, since they conflict with the assumption of monotonic belief in §4.2.



(a) Publication **(b)** Email

(c) Mobile

Fig. 3 Learning curves in terms of F1-score

Table 4 Average F1-score by all selection strategies (%)

Data set	Random	MU	ID	BMS	IMS
Publication	60.6	63.7	64.8	66.4	**66.8**
Email	85.6	86.2	87.3	**87.6**	86.3
Mobile	79.2	80.0	74.3	**80.4**	79.9

The results were obtained by randomly selecting 10 relationships as the initial labeled set Y^L, and then iteratively perform the active selection algorithms, each time with $b = 10$ relationships to query
Bold values indicate the best performance

Information Density (ID): It chooses b nodes with the maximum average similarity to all other nodes in Y^U, proposed in Settles and Craven (2008).

Effect of active learning We plot the learning curves on each data set in Fig. 3, and list the average F1-score by all selection strategies in Table 4. The results clearly demonstrate the effectiveness of the active selection strategies. In the Publication data set, the overall F1-score of the IMS strategy with 100 samples labeled outperforms the Random algorithm by +7.4 %. In Email and Mobile data set, the BMS strategy achieves the best performance, with an improvement of +3.3 and +3.9 % respectively.

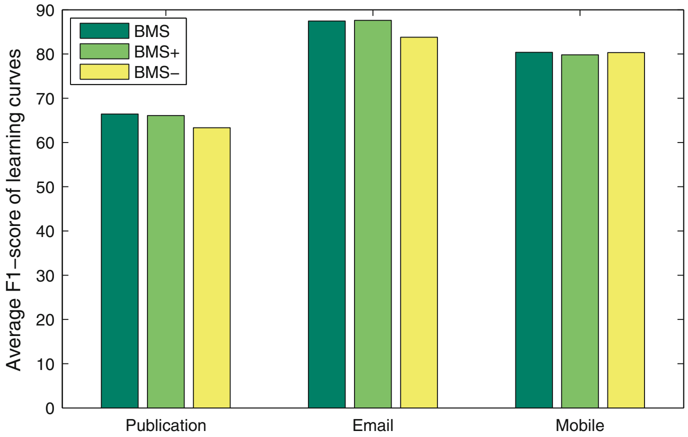


Fig. 4 Performance comparison between variations of BMS

Performance comparison In the publication data set, both the proposed BMS and IMS strategy significantly perform better than all the baseline methods (paired t tests with 95 % significance). BMS also significantly outperforms Random and ID strategy in Mobile data set, while its performance is close to MU. The performance of IMS is shown better than ID in Mobile data set, but is close to other baseline methods. In Email data set, BMS significantly outperforms Random, while the performance of other methods seems close to each other. Generally, the proposed BMS strategy performs more consistently, and obtain better result in two of the three data sets. The performance of IMS strategy is the best in Publication data set, but seems close to baseline methods in the other two data sets.

Network information helps According to factor contribution analysis mentioned before, co-advisor factor in Publication data set contributes the most. This explains why the proposed methods achieve better performance than the alternative baseline methods in Publication data set. The average F1-score of BMS and IMS reaches 65 % with less than 30 labeled samples, while ID uses more than 40, and MU uses more than 60. In Email and Mobile data set, BMS still takes advantage of the network information, but the improvement shrinks due to the considerable decrease of factor contribution.

In-depth analysis of BMS There are also some variations of BMS and we conduct a comparison between them. BMS+ selects all b nodes optimizing $Q_{BMS^+}(A)$, while BMS− employs $Q_{BMS^-}(A)$. Figure 4 shows the average F1-score of the different versions. In Publication and Email data set, the difference between BMS and BMS+ is minor, while the performance of BMS− drops. It might be resulted from different criteria of these three strategies. BMS+ tends to obtain true-positive samples, whereas BMS− is more likely to acquire true-negative samples. F1-score excludes the impact of true-negative samples, and therefore undermines the performance of BMS−. The gap disappears in Mobile data set, probably due to the weak contribution of its correlation and constraint factors.

Table 5 Data transferred in distributed learning algorithm

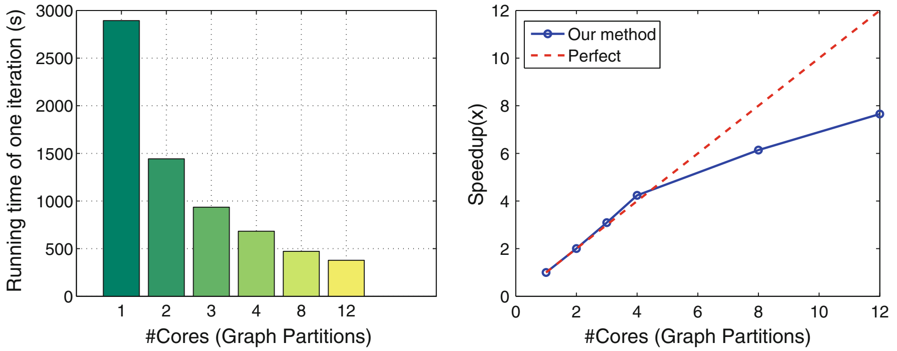
Phase	From	To	Data description
Initialization	Master	Slave i	i-th subgraph
Iteration beginning	Master	Slave i	Current parameters θ
Iteration ending	Slave i	Master	Gradient in i-th subgraph

5.4 Distributed learning

As real social networks may contain millions of users and relationships, it is important for the learning algorithm to scale well with large networks. To address this issue, we develop a distributed learning method based on MPI (Message Passing Interface). The learning algorithm can be viewed as two steps: (1) compute the gradient for each parameter via loopy belief propagation; (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, i.e., one master node is responsible for optimizing parameters, and the other slave nodes are responsible for calculating gradients. At the beginning of the algorithm, the graphical model of PLP-FGM is partitioned into P roughly equal parts, where P is the number of slave processors. This process is accomplished by graph segmentation software METIS (Karypis and Kumar 1998). The subgraphs are then distributed over slave nodes. Note that in our implementation, the edges (factors) between different subgraphs are eliminated, which results in an approximate solution. In each iteration, the master node sends the newest parameters θ to all slaves. Slave nodes then start to perform Loopy Belief Propagation on the corresponding subgraph to calculate the marginal probabilities, then further compute the parameter gradient and send it back to the master. Finally, the master node 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 nodes are summarized in Table 5.

We conduct a series of experiments to evaluate the scalability performance of our distributed learning algorithm on the Publication data set. Figure 5 shows the running time and speedup of the distributed algorithm with different number of computer nodes (2,3,4,8,12 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, it can achieve $\sim 8\times$ speedup with 12 cores. It is noticeable that the speedup curve is beyond the perfect line when using 4 cores, it is not strange since our distributed strategy is approximated. In our distributed implementation, graphs are partitioned into subgraphs, and the factors across different parts are discarded. Thus, the graph processed in distributed version contains less edges, making the computational cost less than the amount in the original algorithm. The effect of subgraph partition is illustrated in Fig. 6. By using good graph partition algorithm such as METIS, the performance only decreases slightly (1.4 % in accuracy and 1.6 % in F1-score). A theoretical study of



(a) Running time vs. #cores **(b)** Speedup vs. #cores

Fig. 5 Scalability performance

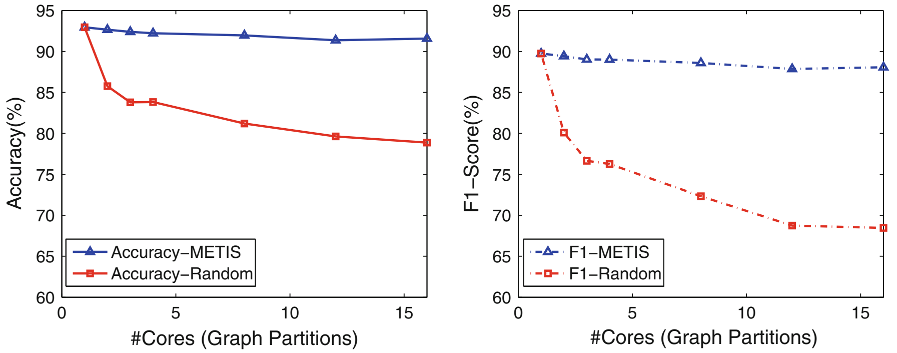


Fig. 6 Approximation of graph partition

the approximate ratio for the distributed learning algorithm would be an interesting issue and is also one of our ongoing work.

6 Related work

Relationship mining Relationship mining is an important problem in social network analysis. One research branch is to predict and recommend unknown links in social networks. Liben-Nowell and Kleinberg (2007) study the unsupervised methods for link prediction. Menon and Elkan (2010) propose a log-linear matrix model for dyadic prediction. Backstrom and Leskovec (2011) propose a supervised random walk algorithm to estimate the strength of social links. Leskovec et al. (2010) employ a logistic regression model to predict positive and negative links in online social networks, where the positive links indicates the relationships such as friendship, while negative indicating opposition. However, these works consider only the black-white social networks, and do not consider the types of the relationships. There are also several works on mining the relationship semantics. Diehl et al. (2007) try to identify the manager-subordinate relationships by learning a ranking function. Wang et al. (2010) propose an unsupervised probabilistic model for mining the advisor-advisee relationships from the publication network. Eagle et al. (2009) present several patterns discovered in mobile phone data, and try to use these pattern to infer the friendship network. Tang and Liu (2011) develop a classification framework of social media based on differentiating different types of social connections. However, these algorithms mainly focus on a specific domain, while our model is general and can be applied to different domains. Moreover, these methods do not explicitly consider the correlation information between different relationships. Hopcroft et al. (2011) explore the problem of reciprocal relationship prediction and Tang et al. (2012) have developed a framework for classifying the type of social relationships by learning across heterogeneous networks. Tan et al. (2011) have investigated how different types of relationships between users influence the change of users' opinion. However, they do not consider how to make optimal use of user interaction.

Another related research topic is relational learning (Califf and Mooney 1999; Getoor and Taskar 2007). However, the problem presented in this paper is very different. Relational learning focuses on the classification problems when objects or entities are presented in relations, while this paper explores the relationship types in social network. A number of supervised methods for link prediction in relational data have also been developed (Taskar et al. 2003; Popescul and Ungar 2003).

In our previous work Tang et al. (2011), we study the problem of inferring social ties in large networks. In this work, we extend the work by further studying active learning for inferring social ties. We propose two active learning models, the Influence-maximization Selection (IMS) model and the Belief-maximization Selection (BMS) model. We give a theoretical analysis to the BMS model, which guarantee to have a bounded approximation ratio using a greedy strategy. Our exploration shows that the accuracy of inferring social ties can be improved by using active learning.

Active learning for networked data The exploding data size in social network, bioinformatics, and many other fields nourishes the development of studies on active learning of networked data. Settles and Craven (2008) provide a survey for active learning strategies in sequence labeling tasks. There have been several works focus on some special graphical structures. Krause and Guestrin (2009) develop an algorithm to non-myopically optimize the active learning task in chain graphical models. Cesa-Bianchi et al. (2010) study active learning on trees. Also, active learning on several specific model has been explored. Martinez and Tsechpenakis (2008) combine active learning method with a CRF-based classifier and propose an automated online learning framework. Golovin et al. (2010) develop a greedy algorithm for Bayesian active learning with noise. Shi et al. (2011) study the problem of batch mode active learning for networked data. They propose a batch mode active learning method by combining three criteria (i.e., minimum redundancy, maximum uncertainty and maximum impact). Similar to our intuition, Kuwadekar and Neville (2011) also attempt to leverage the network structure to benefit the active learning performance. But they study the problem on across-network task, while we focus on within-network setting. Kimura et al. (2010) propose an effective algorithm to extract influential nodes in a social network. In addition, Bilgic et el. (2010) propose a method for active inference on networked data, based on uncertainty sampling, committee-based sampling and clustering; Roy and McCallum (2001) directly optimize expected future error.

7 Conclusion

In this paper, we study the problem of inferring social ties in large networks. We define the problem in a semi-supervised framework, and propose a Partially-Labeled Pairwise Factor Graph Model (PLP-FGM) for learning to infer the relationship semantics. In PLP-FGM, relationships in social network are modeled as nodes, and the type of social relationships are modeled as hidden variables. An efficient algorithm is presented to learn model parameters and to predict unknown relationships. Experimental results on three different types of data sets validate the effectiveness of the proposed model. To further scale up to large networks, a distributed learning algorithm has been developed. Experiments demonstrate good parallel efficiency of the distributed learning algorithm. In order to effectively learn the mapping function, we propose two active learning strategies: Influence-Maximization Selection and Belief-Maximization Selection, both aiming to capture the inter-relationship influence. Experimental results show that BMS and IMS often achieve significant better performance than baseline methods.

So far PLP-FGM is utilized to infer social ties in a static social network. However, most social network in our life is dynamic and it would be worthwhile to extend the model to the dynamic case. PLP-FGM works in a semi-supervised framework, requiring labeled data to learn the predictive function. Some unsupervised methods can be studied for the social tie inferring problem for a broader utilization.

Inferring social ties represents a new research direction in social network analysis. As future work, it is interesting to further study to which extent we should trust user feedbacks. It would also be interesting to investigate how the inferred relationship semantics can help other applications such as community detection, influence analysis, and link recommendation.

Acknowledgments The work is supported by the Natural Science Foundation of China (No. 61073073, No. 60973102, No. 61170061),Chinese National Key Foundation Research (No. 60933013, No. 61035004), and National Basic Research Program of China (No. 2011CB302302). It is also supported by a research funding from Nokia Research Center.

Appendix: Feature definition

In this section, we introduce how we define the attribute factor functions. In the Publication data set, we define five categories of attribute factors: Paper count, Paper ratio, Coauthor ratio, Conference coverage, First-paper-year-diff. The definitions of the attributes are summarized in Table 6. In the Email data set, traffic-based features are extracted. For a relationship, we compute the number of emails for different communication types. In the Mobile data set, the attributes we extracted are #voice calls, #messages, Night-call ratio, Call duration, #proximity and In-role proximity ratio.

Table 6 Attributes used in the experiments

Data set	Factor	Description	
Publication	Paper count	$\|P_i\|,\ \|P_j\|$	
	Paper ratio	$\|P_i\|/\|P_j\|$	
	Coauthor ratio	$\|P_i \cap P_j\|/\|P_i\|,\ \|P_i \cap P_j\|/\|P_j\|$	
	Conference coverage	The proportion of the conferences which both v_i and v_j attended among conferences v_j attended	
	First-paper-year-diff	The difference in year of the earliest publication of v_i and v_j.	
Email	Traffics	Sender	Recipients Include
		v_i	v_j
		v_j	v_i
		v_i	v_k and not v_j
		v_j	v_k and not v_i
		v_k	v_i and not v_j
		v_k	v_j and not v_i
		v_k	v_i and v_j
Mobile	#voice calls	The total number of voice call logs between two users	
	#messages	Number of messages between two users	
	Night-call ratio	The proportion of calls at night (8 PM to 8 AM)	
	Call duration	The total duration time of calls between two users	
	#proximity	The total number of proximity logs between two users	
	In-role proximity ratio	The proportion of proximity logs in "working place" and in working hours (8 AM to 8 PM)	

In the Publication data set, we use P_i and P_j to denote the set of papers published by author v_i and v_j respectively. For a given relationship (v_i, v_j), five categories of attributes are extracted. In the Email data set, for relationship (v_i, v_j), number of emails for different communication types are computed. In the Mobile data set, the attributes are from the voice call/message/proximity logs

References

Albert R, Barabasi AL (2002) Statistical mechanics of complex networks. Rev Mod Phys 74(1), 47–97

Backstrom L, Leskovec J (2011) Supervised random walks: predicting and recommending links in social networks. In: WSDM, pp 635–644

Bilgic M, Mihalkova L, Getoor L (2010) Active learning for networked data. In: Fürnkranz J, Joachims T (eds) ICML. Omnipress, pp 79–86

Califf ME, Mooney RJ (1999) Relational learning of pattern-match rules for information extraction. In: AAAI/IAAI, pp 328–334

Cesa-Bianchi N, Gentile C, Vitale F, Zappella G (2010) Active learning on trees and graphs. In: COLT, pp 320–332

Crandall D, Backstrom L, Cosley D, Suri S, Huttenlocher D, Kleinberg J (2010) Inferring social ties from geographic coincidences. PNAS 107(52):22436

Diehl CP, Namata G, Getoor L (2007) Relationship identification for social network discovery. In: AAAI, AAAI Press, pp 546–552

Domingos P, Richardson M (2001) Mining the network value of customers. In: KDD, pp 57–66

Eagle N, Pentland AS, Lazer D (2008) Mobile phone data for inferring social network structure. In: Social computing, behavioral modeling, and prediction, pp 79–88

Faloutsos M, Faloutsos P, Faloutsos C (1999) On power-law relationships of the internet topology. In: SIGCOMM, pp 251–262

Getoor L, Taskar B (2007) Introduction to statistical relational learning. The MIT Press, Cambridge

Golovin D, Krause A, Ray D (2010) Near-optimal Bayesian active learning with noisy observations. CoRR abs/1010.3091

Grob R, Kuhn M, Wattenhofer R, Wirz M (2009) Cluestr: mobile social networking for enhanced group communication. In: GROUP, pp 81–90

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

Hopcroft JE, Lou T, Tang J (2011) Who will follow you back? Reciprocal relationship prediction. In: CIKM'11

Karypis G, Kumar V (1998) MeTis: unstrctured graph partitioning and sparse matrix ordering system. Version 4.0 Sept

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: KDD, pp 137–146

Kimura M, Saito K, Nakano R, Motoda H (2010) Extracting influential nodes on a social network for information diffusion. Data Min Knowl Discov 20(1):70–97

Kleinberg J (2005) Temporal dynamics of on-line information streams. In: Garofalakis M, Gehrke J, Rastogi R (eds) Data stream managemnt processing high-speed data. Springer, Heidelberg

Krause A, Guestrin C (2009) Optimal value of information in graphical models. J Artif Intell Res (JAIR) 35:557–591

Kuwadekar A, Neville J (2011) Relational active learning for joint collective classification models. In: Getoor L, Scheffer T (eds) Proceedings of the 28th International Conference on Machine Learning (ICML-11), ICML '11, pp 385–392, New York, NY, USA, June. ACM.

Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th international conference on machine learning (ICML'01), pp 282–289

Leskovec J, Huttenlocher DP, Kleinberg JM (2010) Predicting positive and negative links in online social networks. In: WWW, pp 641–650

Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. J Am Soc Inf Sci Technol 58(7):1019–1031

Martinez O, Tsechpenakis G (2008) Integration of active learning in a collaborative CRF. In: Computer vision and pattern recognition workshop, pp 1–8

Menon AK, Elkan C (2010) A log-linear model with latent features for dyadic prediction. In: ICDM, pp 364–373

Murphy K, Weiss Y, Jordan M (1999) Loopy belief propagation for approximate inference: an empirical study. In: UAI, vol 9, pp 467–475

Newman MEJ (2003) The structure and function of complex networks. SIAM Rev 45:167–256

Popescul A, Ungar L (2003) Statistical relational learning for link prediction. In: IJCAI03 workshop on learning statistical models from relational data volume 149,172

Roth M, Ben-David A, Deutscher D, Flysher G, Horn I, Leichtberg A, Leiser N, Matias Y, Merom R (2010) Suggesting friends using the implicit social graph. In: KDD, pp 233–242

Roy N, McCallum A (2001) Toward optimal active learning through sampling estimation of error reduction. In: ICML, pp 441–448

Settles B, Craven M (2008) An analysis of active learning strategies for sequence labeling tasks. In: EMNLP, pp 1070–1079

Shi L, Zhao Y, Tang J (2011) Batch mode active learning for networked data. In: ACM Transactions on Intelligent Systems and Technology (TIST)

Strogatz SH (2003) Exploring complex networks. Nature 410:268–276

Tan C, Lee L, Tang J, Jiang L, Zhou M, Li P (2011) User-level sentiment analysis incorporating social networks. In: KDD, pp 1397–1405

Tan C, Tang J, Sun J, Lin Q, Wang F (2010) Social action tracking via noise tolerant time-varying factor graphs. In: KDD, pp 1049–1058

Tang J, Lou T, Kleinberg J (2012) Inferring social ties across heterogenous networks. In: WSDM'12

Tang J, Sun J, Wang C, Yang Z (2009) Social influence analysis in large-scale networks. In: KDD, pp 807–816

Tang J, Zhang J, Yao L, Li J, Zhang L, Su Z (2008) Arnetminer: Extraction and mining of academic social networks. In: KDD'08, pp 990–998

Tang L, Liu H (2011) Leveraging social media networks for classification. Data Min Knowl Discov 23(3):447–478

Tang W, Zhuang H, Tang J (2011) Learning to infer social ties in large networks. In: ECML/PKDD'11, pp 381–397

Taskar B, Wong MF, Abbeel P, Koller D (2003) Link prediction in relational data. In: NIPS. MIT Press

Wang C, Han J, Jia Y, Tang J, Zhang D, Yu Y, Guo J (2010) Mining advisor-advisee relationships from research publication networks. In: KDD, pp 203–212

Wang D, Pedreschi D, Song C, Giannotti F, Barabási A-L (2011) Human mobility, social ties, and link prediction. In: KDD, pp 1100–1108

Yang Z, Guo J, Cai K, Tang J, Li J, Zhang L, Su Z (2010) Understanding retweeting behaviors in social networks. In: CIKM, pp 1633–1636