Learning to Infer Social Ties in Large Networks*

Wenbin Tang, Honglei Zhuang, and Jie Tang

Department of Computer Science and Technology, Tsinghua University
{tangwb06,honglei.zhuang}@gmail.com, jietang@tsinghua.edu.cn

Abstract. In online social networks, most relationships are lack of meaning labels (e.g., "colleague" and "intimate friends"), simply because users do not take the time to label them. An interesting question is: can we automatically infer the type of social relationships in a large network? what are the fundamental factors that imply the type of social relationships? In this work, we formalize the problem of social relationship learning into a semi-supervised framework, and propose a Partially-labeled Pairwise Factor Graph Model (PLP-FGM) for learning to infer the type of social ties. We tested the model on three different genres of 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). Finally, we develop a distributed learning algorithm to scale up the model to real large networks.

1 Introduction

With the success of many large-scale online social networks, such as Facebook, MySpace, and Twitter, and the rapid growth of mobile social networks such as FourSquare, online social network has become a bridge between our real daily life and the virtual web space. Facebook, one of the largest social networks, has more than 600 million active users in Jan 2011; Foursquare, a location-based mobile social network, has attracted 6 million registered users by the end of 2010. 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 [1, 7, 18, 21], dynamic evolution analysis [13], social influence analysis [5, 12, 23], and social behavior analysis [20, 22]. However, most of these works ignore one important fact that makes the online social networks very 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

* The work is supported by the Natural Science Foundation of China (No. 61073073, No. 60973102), Chinese National Key Foundation Research (No. 60933013, No.61035004).

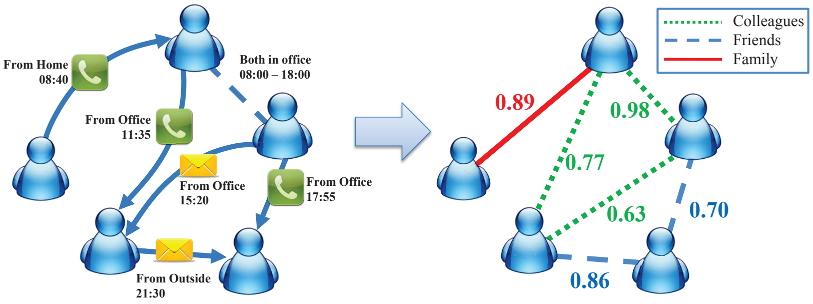


Fig. 1. An example of relationship mining in mobile communication network. The left figure is the input of our problem, and the right figure is the objective of the relationship mining task.

have created custom contact groups [20, 10] and less than 23% connections on LinkedIn have been labeled. Identification of the type of social relationships can benefit many applications. For example, if we could have extracted friendships between users from the mobile communication network, we can leverage the friendships for a "word-of-mouth" promotion of a new product [12].

In this work, we investigate to what extent social relationships can be inferred from the online social networks: E.g., given users' behavior history and interactions between users, can we estimate how likely they are to be family members? There exist a few related studies. For example, Diehl et al. [4] try to identify the relationships by learning a ranking function. Wang et al. [26] propose an unsupervised algorithm for mining the advisor-advisee relationships from the publication network. However, both algorithms focus on a specific domain (Email network in [4] and Publication network in [26]) and are not easy to extend to other domains. It is well recognized that the type of users' relationships in a social network can be implied by various complex and subtle factors [9, 14]. One challenging question is: can we design a unified model so that it can be easily applied to different domains?

Motivating Examples To illustrate the problem, Figure 1 gives an example of relationship mining in mobile calling network. The left figure is the input of our problem: a mobile social network, which consists of users, calls and messages between users, and users' location logs, etc. Our objective is to infer the type of the relationships in the network. In the right figure, the users who are family members are connected with a red-colored line, friends are connected with a blue-colored dash line, and colleagues are connected with a green-colored dotted line. The probability associated with each relationship represents our confidence on the detected relationship types.

Thus, the problem becomes how to design a flexible model for effectively and efficiently mining relationship types in different networks. This problem is nontrivial and poses a set of unique challenges. First, what are the underlying factors that may determine 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. 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.

Contributions In this paper, we try to conduct a systematic investigation of the problem of inferring social relationship types in large networks with the following contributions:

- We formally formulate the problem of inferring social relationship in large networks, and propose a partially-labeled pairwise factor graph model (PLP-FGM).
- We present a distributed implementation of the learning algorithm based on MPI (Message-Passing Interface) to scale up to large networks.
- We conduct experiments on three different data sets: Publication, Email, Mobile network. Experimental results show that the proposed PLP-FGM model can be applied to the different scenarios and clearly achieves better performance than several alternative models.

The rest of paper is organized as follows. Section 2 formally formulates the problem; Section 3 explains the PLP-FGM model; Section 4 gives experimental results; Finally, Section 5 discusses related work and Section 6 concludes.

2 Problem Definition

In this section, we first give several necessary definitions and then present the problem formulation.

A social network can be represented as $G = (V, E)$, where V is the set of $|V| = N$ users and $E \subset V \times V$ is the 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. To begin with, let us first give a formal definition of the output of the 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, and* 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 calling 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 change 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. 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 already exist some labeled relationships. Formally, we can define the input of our problem, 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; 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} denotes 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

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

Our formulation of inferring social relationships is very different from existing works on relation mining [3]. They focus on detecting the relationships from the content information, while we focus on mining relationship semantics in social networks. Both Diehl et al.[4] and Wang et al.[26] investigate the problem of relationship identification. However, they focus on the problem in specific domains (Email network or Publication network).

3 Partially-Labeled Pairwise Factor Graph Model (PLP-FGM)

3.1 Basic Idea

In general, there are two ways to model the problem. The first way is to model each user as a node and for each node we try to estimate probability distributions of different relationships from the user to her neighborhood nodes in the social network. The graphical model 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 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

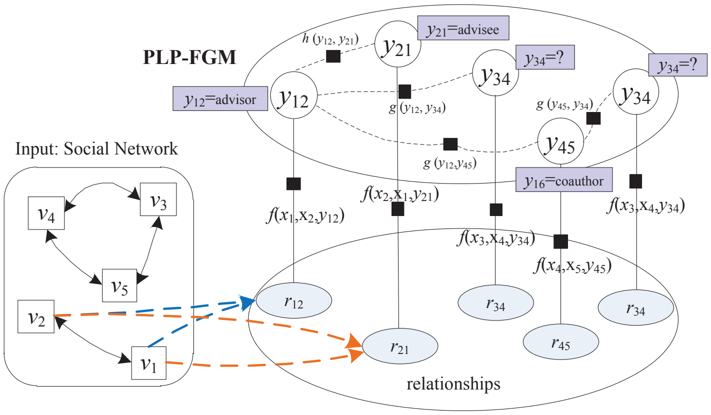


Fig. 2. Graphical representation of the PLP-FGM model.

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). More importantly, this model is able to incorporate different correlations between relationships.

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 made 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 of 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 need also consider some global constraints such as common knowledge or user-specific constraints.

3.2 Partially-Labeled Pairwise Factor Graph Model (PLP-FGM)

Based on 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 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 detail. 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

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

where λ is a weighting vector and $\mathbf{\Phi}$ 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.

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 $\mathbf{s}(y_i) = (\mathbf{\Phi}(y_i, \mathbf{x}_i)^T, \sum_{y_j} \mathbf{g}(y_i, y_j)^T, \sum_{y_j} \mathbf{h}(y_i, y_j)^T)^T$. The joint probability defined in (Eq. 1) can be written as

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

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 the 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 $\mathcal{O}(\theta)$:

```
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.

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

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

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

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 second expectation $\mathbb{E}_{p_\theta(Y|G)\mathbf{S}}$. A number of approximate algorithms have been proposed, such as Loopy Belief Propagation (LBP) [17] and Mean-field [28]. 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 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 largest marginal probability. The marginal probability is then taken as the prediction confidence.

3.3 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 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[11]. 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, but very efficient 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 1.

4 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 differ-

Table 1. 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

Table 2. 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

ent genres of data sets to evaluate the effectiveness and efficiency of our proposed approach. All data sets and codes are publicly available.[1]

4.1 Experiment Setup

Data sets. We perform our experiments on three different data sets: Publication, Email, and Mobile. Statistics of the data sets are shown in Table 2.

- Publication. In the publication data set, we try to infer the advisor-advisee relationship from the coauthor network. The data set is provided by [26]. Specifically, we have collected 1,632,442 publications from Arnetminer [24] (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[2]; 3) extracted from AI Genealogy project[3]. 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 [4].
- Mobile. In the mobile data set, we try to infer the friendship in mobile calling network. The data set is from Eagle et al. in [6]. 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.

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

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

[3] `http://aigp.eecs.umich.edu`

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 (v_i, v_j), our objective 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 possible 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 one, 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 in 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 features in the three data sets. A detailed description of the factor definition for each data set is given in Table 5 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 [26] 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. Another 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.

Table 3. 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**

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.

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).

4.2 Accuracy Performance

Table 3 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 would hurt the precision. As a result, TPFG underperforms our method 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 clearly showed 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.

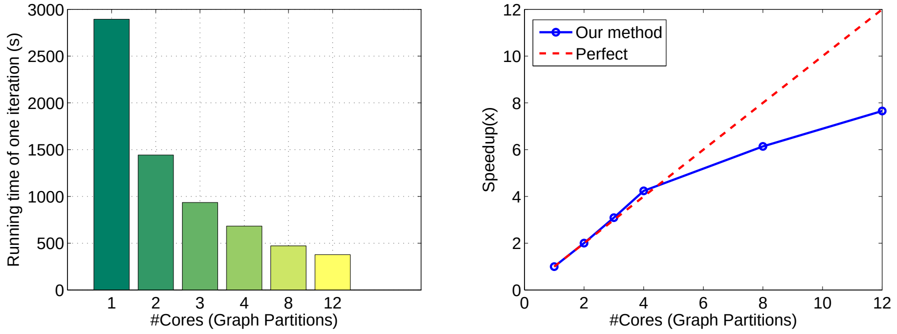
Table 4. Factor contribution analysis on three data sets. (%).

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%)

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 4 shows the 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. However, 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. This is because our model not only considers different factors, but also leverages the correlation between them.

4.3 Scalability Performance

We now evaluate the scalability performance of our distributed learning algorithm on the Publication data set. Figure 3 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



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

Fig. 3. Scalability performance.

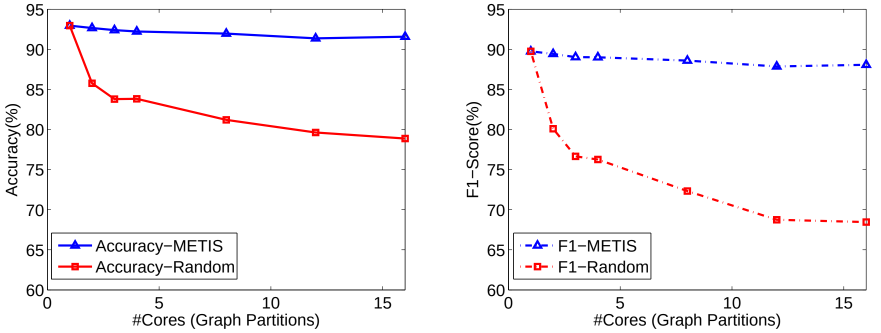


Fig. 4. Approximation of graph partition.

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 Figure 4. 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 the approximate ratio for the distributed learning algorithm would be an interesting issue and is also one of our ongoing work.

5 Related work

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 et al.[16] study the unsupervised methods for link prediction. Xiang et al. [27] develop a latent variable model to estimate relationship strength from interaction activity and user similarity. Backstrom et al. [2] propose a supervised random walk algorithm to estimate the strength of social links. Leskovec et al. [15] 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. [4] try to identify the manager-subordinate relationships by learning a ranking function. Wang et al. [26] propose an unsupervised probabilistic model for mining the advisor-advisee relationships from the publication network. Eagle et al. [6] present several patterns discovered in mobile phone data, and try to use these pattern to infer the friendship network. 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.

Another related research topic is relational learning[3, 8]. 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 [25, 19].

6 Conclusion

In this paper, we study the problem of inferring the type of social ties in large networks. We formally define the problem in a semi-supervised framework, and propose a partially-labeled pairwise factor graph model (PLP-FGM) to learn to infer the relationship semantics. In PLP-FGM, relationships in social network are modeled as nodes, the attributes, correlations and global constraints are modeled as factors. An efficient algorithm is proposed 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 is developed. Experiments demonstrate good parallel efficiency of the distributed learning algorithm.

Detecting the relationship semantics makes online social networks colorful and closer to our real physical networks. It represents a new research direction in social network analysis. As future work, it is interesting to study how to further improve the mining performance by involving users into the learning process (e.g., via active learning). In addition, it would be also interesting to investigate how the inferred relationship semantic information can help other applications such as community detection, influence analysis, and link recommendation.

References

1. R. Albert and A. L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1), 2002.

2. L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.
3. M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In *AAAI/IAAI*, pages 328–334, 1999.
4. C. P. Diehl, G. Namata, and L. Getoor. Relationship identification for social network discovery. In *AAAI*, pages 546–552. AAAI Press, 2007.
5. P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, pages 57–66, 2001.
6. N. Eagle, A. S. Pentland, and D. Lazer. Mobile phone data for inferring social network structure. *Social Computing, Behavioral Modeling, and Prediction*, pages 79–88, 2008.
7. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, pages 251–262, 1999.
8. L. Getoor and B. Taskar. *Introduction to statistical relational learning*. The MIT Press, 2007.
9. M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.
10. R. Grob, M. K. 0002, R. Wattenhofer, and M. Wirz. Cluestr: mobile social networking for enhanced group communication. In *GROUP*, pages 81–90, 2009.
11. G. Karypis and V. Kumar. *MeTis: Unstrctured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*, Sept. 1998.
12. D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.
13. J. Kleinberg. Temporal dynamics of on-line information streams. In *Data Stream Managemnt: Processing High-speed Data*. Springer, 2005.
14. D. Krackhardt. *The Strength of Strong Ties: The Importance of Philos in Organizations*, pages 216–239. Harvard Business School Press, Boston, MA.
15. J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650, 2010.
16. D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.
17. K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, volume 9, pages 467–475, 1999.
18. M. E. J. Newman. The structure and function of complex networks. *SIAM Reviews*, 45, 2003.
19. A. Popescul and L. Ungar. Statistical relational learning for link prediction. In *IJCAI03 Workshop on Learning Statistical Models from Relational Data*, volume 149, page 172, 2003.
20. M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. In *KDD*, pages 233–242, 2010.
21. S. H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2003.
22. C. Tan, J. Tang, J. Sun, Q. Lin, and F. Wang. Social action tracking via noise tolerant time-varying factor graphs. In *KDD*, pages 1049–1058, 2010.
23. J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, pages 807–816, 2009.
24. J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.
25. B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *NIPS*. MIT Press, 2003.

26. C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo. Mining advisor-advisee relationships from research publication networks. In *KDD*, pages 203–212, 2010.
27. R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *WWW*, pages 981–990, 2010.
28. E. P. Xing, M. I. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *UAI'03*, pages 583–591, 2003.

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 5. 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 5. Attributes used in the experiments. 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.

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 (8pm to 8am).	
	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 (8am to 8pm).	