Wan HY, Lin YF, Wu ZH *et al.* Discovering typed communities in mobile social networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 27(3): 480–491 May 2012. DOI 10.1007/s11390-012-1237-9

Discovering Typed Communities in Mobile Social Networks

Huai-Yu Wan (万怀宇), *Student Member*, *CCF*, You-Fang Lin (林友芳), Zhi-Hao Wu (武志昊), and Hou-Kuan Huang (黄厚宽), *Senior Member*, *CCF*

School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

E-mail: {huaiyuwan, yflin, zhihaowu, hkhuang}@bjtu.edu.cn

Received August 31, 2011; revised February 15, 2012.

Abstract Mobile social networks, which consist of mobile users who communicate with each other using cell phones, are reflections of people's interactions in social lives. Discovering typed communities (e.g., family communities or corporate communities) in mobile social networks is a very promising problem. For example, it can help mobile operators to determine the target users for precision marketing. In this paper we propose discovering typed communities in mobile social networks by utilizing the labels of relationships between users. We use the user logs stored by mobile operators, including communication and user movement records, to collectively label all the relationships in a network, by employing an undirected probabilistic graphical model, i.e., conditional random fields. Then we use two methods to discover typed communities based on the results of relationship labeling: one is simply retaining or cutting relationships according to their labels, and the other is using sophisticated weighted community detection algorithms. The experimental results show that our proposed framework performs well in terms of the accuracy of typed community detection in mobile social networks.

Keywords mobile social network, typed community detection, relationship labeling, conditional random field

1 Introduction

Social networks are a ubiquitous paradigm of human interactions in real world. A social network is a set of nodes (people, families, organizations or other social entities) connected by a set of relationships with different types (e.g., family, friendship, co-working, collaboration, communication). In recent years, as the proliferation of online social networks and mobile social networks, social network analysis (SNA)[1-2] has attracted an increasing interest in the research community, which greatly promotes our understanding of the important structural patterns of the interactions between people.

In this paper, we focus on mobile social networks[3-5], a particular kind of social network which consists of mobile users who communicate with one another using cell phones. Compared to other electronic social networks, e.g., email networks and online social networks, mobile social networks reflect people's the most realistic interactions in social lives, because mobile phone calls are comparatively more common than Internet-based communications in everyday lives. And the widespread use of mobile phones offers unprecedented opportunities to construct maps of society-wide communication networks. For instance, as of April 2011, the number of mobile users in China has exceeded 900 millions[6]. In addition, mobile social networks contain the users' sufficient location information as well as their communication logs, which is essential for us to uncover the properties and structures of large-scale social networks.

As the competition in the mobile communication domain is very fierce, attracting and retaining users becomes a strategic challenge for mobile operators. Except for the traditional mass marketing strategies which are directed to all mobile users, targeted and personalized precision marketing becomes one of the most effective ways to attract users, such as customizing special services for particular user groups. In order to determine the target users for precision marketing, we need to discovery typed communities (e.g., family communities or corporate communities) in mobile social networks. Mobile users in such communities are connected by a single type of relationships (such as family or colleague) and have similar behavior and consumption patterns. Once such typed communities are discovered, mobile operators can customize or recommend appropriate services or products for them according to their particular characteristics, and thus promote the quality of mobile services and further retain their users.

Regular Paper

This work was partially supported by the Fundamental Research Funds for the Central Universities of China, the National Natural Science Foundation of China under Grant No. 60905029, and the Beijing Natural Science Foundation under Grant No. 4112046.

Community structure[7-8] is one of the most important properties of mobile social networks. Many researches on mobile social networks, such as the study of message spreading[9] and mining top-K influential users[4], are based on the community structure of networks. A general community detection problem can be modeled as an unsupervised learning problem, and a large number of related algorithms have been developed in recent years. However, general community detection algorithms cannot generate typed communities because they do not take into account their types. Actually, in social networks, the types (i.e., the topics) of communities are determined by the labels of relationships between people. For example, family relationships form family communities, while friend relationships form friend communities. Fig.1 depicts a simple example, in which the solid lines indicate the family relationships while the dashed lines indicate the friend relationships.

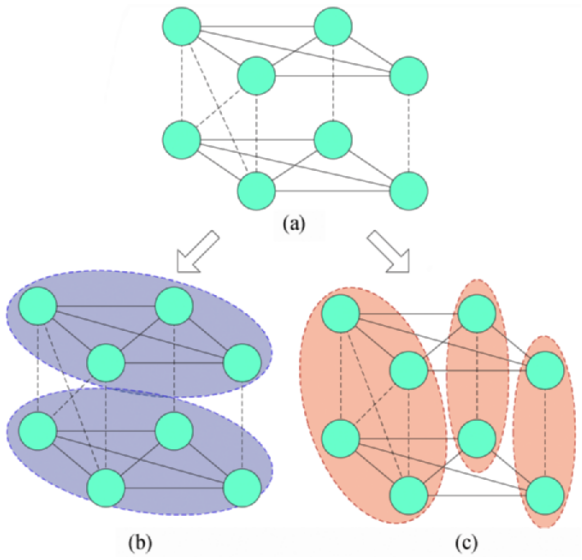

Fig.1. Simple example of typed communities in a social network. (a) Example of a social network. (b) Family community structure. (c) Friend community structure.

In this paper, we discuss how to discovery typed communities in mobile social networks. Practically, if we can identify the types of relationships between mobile users, it is possible for us to discover typed communities. As we know, the user logs stored by mobile operators record huge amounts of user behavior data, including call detail records (CDRs), short message (SM) data, user movement information, etc. CDRs are the call logs between mobile users, and each call log consists of calling user, called user, call date, call time, call duration, etc. SM data is the SM logs between mobiles users and each SM log consists of sender, receiver, date of message, time of message, content of message, etc. User movement information is the records of cell switching of users when some events (e.g., powering on/off, calling, moving from cell to cell) occurred. Each record consists of user, event type, event date, event time, cell ID, etc.

From this user behavior information, we can infer some communication and movement patterns between mobile users. In general, such patterns are very different for relationships with different types. For instance, a typical movement pattern in a workday is that two mobile users with a family relationship tend to be at the same locations in off hours but at different locations in working hours (see Fig.2(a)), while the situation for two users with a colleague relationship is just on the contrary (see Fig.2(b)). And a typical communication pattern in a workday is that two users with a family relationship tend to communicate more closely in off hours than in working hours, while the situation for two users with a colleague relationship is just on the contrary (as shown in Fig.3).

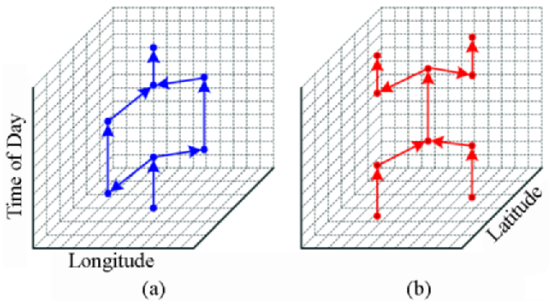

Fig.2. Typical movement pattern in a workday for two mobile users with (a) Family relationship and (b) Colleague relationship.

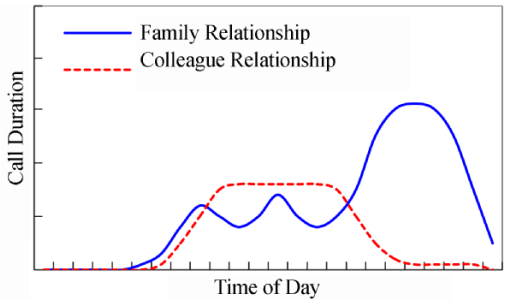

Fig.3. Typical communication pattern in a workday for two mobile users with family relationship and colleague relationship respectively.

According to these communication and movement patterns of mobile users, we can infer the types of relationships between them (i.e., relationship labeling) by using various classifiers. Either supervised or semi-supervised learning can be used: supervised learning is suitable for the case that the dataset is naturally split into disjoint subnetworks and some of them are used for

training while the others for test; and semi-supervised learning is suitable for the case that the dataset is a partially labeled within-network in which the known labels are used to estimate the unknown labels. In this paper, considering mobile social networks are spatio-temporal dynamic networks and can be easily split into disjoint components, we just focus on the first situation where supervised learning is employed in our framework. Traditional classifiers, such as naïve Bayes, logistic regression or support vector machine (SVM), assume that all the labels are independent and identically distributed (IID) and classify each relationship separately by only using its content. However, in social networks in practice, the relationship labels are not completely independent. In order to describe the dependencies among the labels accurately, we adopt a probabilistic graphical model, i.e., conditional random fields[10], to simultaneously decide on the class labels of all the relationships in a social network together. Previous research has indicated that this collective classification method can improve the classification accuracy in relational data[11].

Then we can discover typed communities based on the results of relationship labeling. There are two methods: the first method is retaining or cutting relationships according to their labels, and the second method is taking the probabilities of labels as the weights of relationships and then using weighted community detection algorithms to discover typed communities. The disadvantage of the first method is that the community detection accuracy is completely dependent on the prediction accuracy of relationship labeling, and the mislabeling of some key relationships may lead to some serious errors in typed community detection. While the second method can amend some inaccuracies of relationship labeling by utilizing the link structure of a mobile social network.

We conduct our experiments on some real-world datasets. The experimental results show that, classifying the relationships between mobile users by using their communication and movement information and then discovering the typed communities based on the results of relationship labeling by employing some weighted community detection algorithms performs well in terms of accuracy and is a very effective method for typed community detection in mobile social networks.

It should be noted that, in our research, we use the detailed communication and cell switching information which is the privacy of mobile users and must be seriously protected. First, all the phone numbers appearing in our experimental dataset we get from a mobile operator are encrypted irreversibly, so that one cannot backtrack a specific user from the encrypted data. Second, all the cell location information (i.e., the latitude and longitude coordinates) is mapped to a normalized coordinate system, so that we can only get the relative coordinate of a cell but not know the real location of a user. In these ways, we ensure that all users' privacy will not be violated.

The rest of the paper is organized as follows. The next section provides a brief introduction of data preprocessing. Section 3 discusses the relationship labeling problem and Section 4 presents typed community detection methods, followed by the experimental evaluations in Section 5. Finally, we give the conclusions of our work in Section 6.

2 Data Preprocessing

The user logs stored by mobile operators are event-driven data. Each tuple records an event such as calling, sending SM, powering on/off, cell switching. These data are very detailed and large-scale but cannot be used directly for relationship labeling. For obtaining communication and movement patterns between mobile users from these data, we must transform them into a time-driven mode.

We first give two definitions.

Definition 1 (Period of Time). *We divide the* 24 *hours of a day into* N *periods of time on average, which are denoted by* $T(N) = \{t_1, t_2, \ldots, t_N\}$, *where* $N \geqslant 1$. *The duration of each period of time is* $d(N) = (24 \times 60)/N$ *minutes. In principle,* N *can be infinite, but we let* $N \leqslant 96$ *here, i.e., the minimum duration of each period of time is a quarter of an hour.*

Definition 2 (Stable Point). *Suppose that a mobile user* i *has accessed* M *cells* $c_1, c_2, \ldots, c_M$ *over a period of time* t_n, *according to his/her cell switching records, we can compute the total duration* d_{i,t_n,c_m} *of the user at each cell* c_m. *Then we define the stable point* s_{i,t_n} *of the user* i *over the period of time* t_n *is the cell at which he/she stayed the longest duration, i.e.,* $s_{i,t_n} = \arg\max_{c_m}(d_{i,t_n,c_m})$.

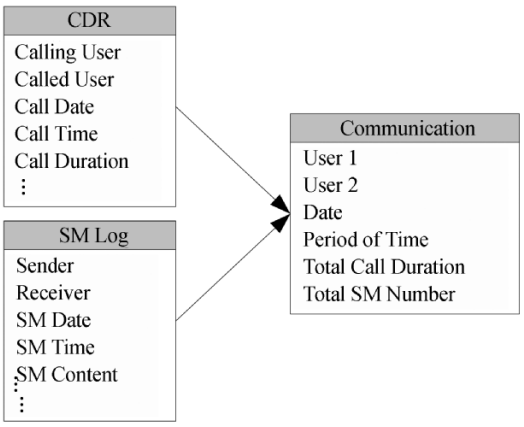

Fig.4. Communication data preprocessing.

Then we can preprocess the user logs based on the above two definitions. For CDRs and SM logs, we compute the total call duration and the total SM number for each pair of communication users over each period of time on each day. And for user movement data, we compute the stable point for each user over each period of time on each day. The transformed time-driven data modes are shown in Figs. 4 and 5 respectively.

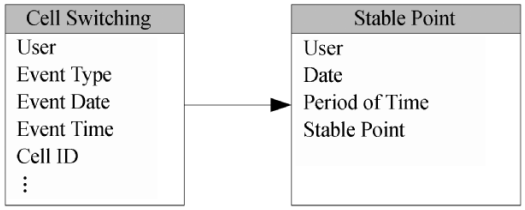

Fig.5. User movement data preprocessing.

3 Relationship Labeling

In this section, we discuss how to identify the types of relationships between mobile users. The goal of relationship labeling is to construct a classification model to label the relationships in mobile social networks by using their communication and movement information. The basic idea for relationship labeling is employing traditional classifiers to classify each relationship separately, where all the labels are assumed to be independent and identically distributed (IID). But in fact, some dependencies exist among the labels of relationships in mobile social networks. For instance, transitive dependencies may exist among the family relationships. Suppose that users 1 and 2 and users 2 and 3 are both family members, then users 1 and 3 should also be family members. Similarly, such transitive dependencies may also exist among the colleague relationships. In order to correctly describe such transitive dependencies among labels in relationship labeling, some relational models should be employed to simultaneously classify all the relationships in a mobile social network together. Previous research has indicated that relational classifiers are usually more accurate than traditional classifiers for the link-based classification problem in social networks[12].

Probabilistic graphical models, including directed models (such as Bayesian networks) and undirected models (such as Markov networks), are often used to construct relational classification system. Undirected graphical models avoid the acyclicity constraint, thus can represent arbitrary forms of dependency between the variables of related instances. In addition, undirected models are well suited for discriminative training, which generally provides significant improvements in classification accuracy over generative training given sufficient training examples[13]. Consequently, in this paper we employ an undirected probabilistic graphical model, *conditional random fields*[10] (or called *conditional Markov networks*, which are extended from Markov networks), to construct our classification system.

Whether we use traditional or relational classifiers, we have to construct features for each relationship in mobile social networks. In this section we first discuss the feature construction problem, and then introduce conditional random fields.

3.1 Feature Construction

We use the preprocessed communication and movement data in the previous section to construct features for each relationship. Usually, people live on a daily basis, so we construct relationship features on a daily cycle. The basic idea is to average the amount of communication and the number of coincide stable points for each period of time.

Generally, the communication and movement patterns of users in workdays are completely different from those in holidays, so we construct features for workdays and holidays separately.

For communication data, because the total amounts of communication vary greatly between different users, the call durations cannot be directly taken as the features and must be normalized, i.e., we divide the total call duration of a period of time by that of the whole day. Then we get the call features as follows:

$$x_{\text{call\_workday}\,t_n} = \frac{1}{N_{\text{w}}} \sum_{k=1}^{N_{\text{w}}} \frac{\text{Call\_Duration}_{k,t_n}}{\text{Call\_Duration}_k},$$

$$x_{\text{call\_holiday}\,t_n} = \frac{1}{N_{\text{h}}} \sum_{k=1}^{N_{\text{h}}} \frac{\text{Call\_Duration}_{k,t_n}}{\text{Call\_Duration}_k},$$

where N_{w} is the number of workdays, N_{h} is the number of holidays, Call_Duration_k, t_n is the total call duration in period of time t_n on day k, and Call_Duration_k is the total call duration on day k.

Similar to the call features, the SM features are:

$$x_{\text{sm\_workday}\,t_n} = \frac{1}{N_{\text{w}}} \sum_{k=1}^{N_{\text{w}}} \frac{\text{SM\_Number}_{k,t_n}}{\text{SM\_Number}_k},$$

$$x_{\text{sm\_holiday}\,t_n} = \frac{1}{N_{\text{h}}} \sum_{k=1}^{N_{\text{h}}} \frac{\text{SM\_Number}_{k,t_n}}{\text{SM\_Number}_k},$$

where $\text{SM\_Number}_{k;t_n}$ is the total SM number in period of time t_n on day k and SM_Number_k is the total SM number on day k. As shown above, the communication data directly reflects the correlations between

users, while a stable point belongs to only a single user, so we need to transform such user attributes to relationship features. Here we simply construct the features by tracking whether the stable points of two users are the same:

$$x_{\text{sp\_workday}\,t_n} = \frac{1}{N_{\text{w}}} \sum_{k=1}^{N_{\text{w}}} \begin{cases} 1, & \text{if } s_{i,k,t_n} = s_{j,k,t_n}, \\ 0, & \text{otherwise}, \end{cases}$$

$$x_{\text{sp\_holiday}\,t_n} = \frac{1}{N_{\text{h}}} \sum_{k=1}^{N_{\text{h}}} \begin{cases} 1, & \text{if } s_{i,k,t_n} = s_{j,k,t_n}, \\ 0, & \text{otherwise}, \end{cases}$$

where s_{i,k,t_n} is the stable point of user i over period of time t_n on day k.

3.2 Conditional Random Fields

3.2.1 Overview

Conditional random fields (CRFs) are undirected graphical models developed for labeling sequence data[10], which represent the conditional distribution over a set of hidden random variables given the observed ones. We use $G = (V, E)$ to represent an undirected graph, where V is the set of nodes in the graph (i.e., a set of discrete random variables) and v is an assignment of values to V. These nodes are connected by a set of undirected edges E which indicates the relevancies between the random variables. We partition V into two subsets: $X \subset V$ is the subset of conditional (or observed) random variables in G, and $Y \subset V$ is the subset of target (or label) random variables, where $X \cup Y = V$ and $X \cap Y \neq \varnothing$. CRFs define a conditional distribution $P(y|x)$ over the hidden states y conditioned on the observations x.

For a graph $G = (V, E)$, a clique c is a set of nodes V_c in G such that each V_i, $V_j \in V_c$ are connected by an edge in E. Cliques play a very important role in the definition of the conditional distribution represented by a CRF. Let $C(G)$ be the set of all cliques in a given graph G. Then, a CRF factorizes the conditional distribution into a product of clique potentials $\phi_c(x_c, y_c)$, where x_c and y_c are the conditional and target variables in clique c respectively. Clique potential ϕ_c is a non-negative real function defined on V_c, which indicates the "compatibility" among the variables in the clique. Given an assignment v_c, the larger the potential value, the more likely the assignment. By using clique potentials, the conditional distribution over the target variables in a graph is defined as:

$$P(y|x) = \frac{1}{Z(x)} \prod_{c \in C(G)} \phi_c(x_c, y_c), \tag{1}$$

where $Z(x)$ is the partition function (also called normalization constant) dependent on x:

$$Z(x) = \sum_{y'} \prod_{c \in C(G)} \phi_c(x_c, y'_c). \tag{2}$$

The potential is often represented by a log-linear combination of a set of feature functions:

$$\begin{aligned} \phi_c(x_c, y_c) &= \exp\Big\{ \sum_k w_k f_k(x_c, y_c) \Big\} \\ &= \exp\{w_c \times f_c(x_c, y_c)\}, \end{aligned} \tag{3}$$

where w_k is the weight of the k-th feature function f_k. Then the log-linear representation of (1) can be abbreviated as follows:

$$\begin{aligned} \log P(y|x) &= \sum_{c \in C(G)} \sum_k w_k f_k(x_c, y_c) - \log Z(x) \\ &= w \times f(x, y) - \log Z(x). \end{aligned} \tag{4}$$

3.2.2 Constructing CRFs

In relationship labeling, we need to make relationships the first-class citizens. The relationships are corresponding to the target variables Y in CRFs and their content attributes are corresponding to the conditional variables X. An edge between any pair of target variables $\langle Y_i, Y_j \rangle$ is established if the relationships i and j are neighboring (i.e., have a common node) in the original mobile social network.

The main task of constructing CRFs is determining the dependencies among the variables, i.e., specifying the cliques and potentials. In relational domain, it is impossible to define cliques for each variable separately, and instead we do this at a template level, where all the cliques in a template share the same potentials and weights.

Two types of cliques need to be defined: *evidence cliques* and *compatibility cliques*. An evidence clique is a dyad clique that consists of a target variable and one of its content attributes. It indicates the direct dependency of the target variable conditioned on the content attribute. Compatibility cliques consist entirely of target variables. In general, we can define dyad compatibility clique templates according to the *encyclopedic regularity* (linked objects tend to have the same class) or the *co-citation regularity* (objects that are cited by the same object tend to have the same class)[14] and triad compatibility clique templates according to the *transitivity regularity* (three objects linked end by end tend to have the same class). However, the encyclopedic regularity and co-citation regularity are not suitable for the relationship labeling problem. Intuitively, in a real-world social network, the two relationships which

have a common person or linked to common relationship may not have any relevance. Consequently, in our model we just define a triad compatibility clique template based on the transitive dependencies among the relationship labels. The principle is very simple: if any three relationships form a loop in the original mobile social network, then we establish a compatibility clique for their corresponding target variables in the CRF.

Fig.6 illustrates a simple example of CRF construction. For simplicity, we hide the content attributes of each relationship in the original mobile social network and the corresponding conditional variables and evidence cliques in the CRF. A dashed ellipse denotes a triad compatibility clique.

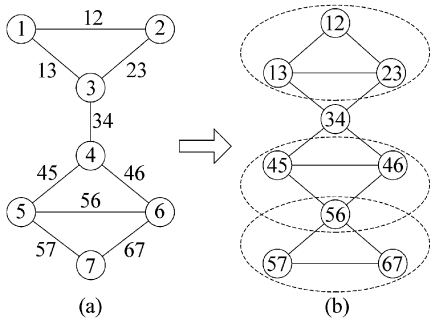

Fig.6. Simple example of CRF construction. (a) Original mobile social network. (b) CRF.

Then we need to define the feature functions for the clique potentials. Here we just define potentials for a binary classification model. For the dyad evidence cliques, we use the indicator functions with the form $f_k(y, x_k) = yx_k$, where $y = \pm 1$ and $x_k \in [0, 1]$. And for the triad compatibility cliques, we simply use a single feature function to track whether the three labels are the same:

$$f_k(y_i, y_j, y_l) = \begin{cases} 1, & \text{if } y_i = y_j = y_l, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

3.2.3 Learning CRFs

The goal of learning a CRF is to determine the parameters (i.e., the weights of the features) in the model. As mentioned previously, CRFs learn these parameters discriminatively, i.e., the weights are determined so as to maximize the conditional distribution $P(y|x)$ of labeled training data. This is in contrast to generative learning, which aims to learn a model the joint probability $P(y, x)$. Maximum a posteriori (MAP) training is used to learn CRFs. To avoid overfitting, we assume the prior of the weights w is a zero-mean Gaussian. A single instantiation $\mathcal{I}$ is used as a training dataset.

• *Maximum Likelihood Estimation* (*MLE*). The log of the MAP objective function based on the likelihood of the training data is as follows:

$$\begin{aligned} L(\mathcal{I}, w) &= \log\Big(P(y|x)\prod_k P(w_k)\Big) \\ &= \log P(y|x) + \sum_k \frac{-w_k^2}{2\sigma^2} - \log\sqrt{2\pi\sigma^2} \\ &= w \times f(x, y) - \log Z(x) - \frac{\|w\|_2^2}{2\sigma^2} + C. \end{aligned} \tag{6}$$

$L(\mathcal{I}, w)$ is a concave function and we can estimate the parameters w by maximizing the objective function by using a variety of gradient-based optimization algorithms, such as conjugate gradient or quasi-Newton. The gradient of the objective function is computed as

$$\nabla L(\mathcal{I}, w) = f(x, y) - E_{P_w}[f(x, Y)]) - \frac{w}{\sigma^2}. \tag{7}$$

From (7) it is seen that the gradient is just the difference between the empirical feature values $f(x, y)$ and the expected feature values $E_{P_w}[f(x, Y)])$, minus a prior term. The expected feature values is related to P_w:

$$E_{P_w}[f(x, Y)] = \sum_{y'} f(x, y')P_w(y'|x). \tag{8}$$

When we use some optimization algorithm to maximize the objective function, we need to compute the partition function $Z(x)$ in (6) and the expected feature counts in (7). The sum over y' involves the exponential number of assignments to all of the target variables, so computing the objective function directly is an NP-hard problem in general. It requires that we run inference in CRFs.

• *Maximum Pseudo-Likelihood Estimation* (*MPLE*). An alternative of MLE is to maximum the *pseudo-likelihood*[15] of the training data. As an efficient alternative of likelihood, the pseudo-likelihood measure is often employed to approximate the joint probability distribution of a collection of random variables with a set of conditional probability distributions (CPDs). Since pseudo-likelihood can only capture the local dependencies and ignores the indirect effects between the non-neighboring variables, it may lose some accuracy in practice. But it can effectively resolve the time complexity problem of relational learning models and makes them much easier to deploy in large-scale social networks[16].

Given a graph $G = (V, E)$, for each label $Y_i \in V$, pseudo-likelihood uses a local CPD $P(y_i|MB(y_i))$ to represent the conditional probability of the label value y_i, where $MB(y_i)$ is the state of the *Markov blanket*

of Y_i in the data. We maximize the following pseudo-likelihood

$$P(y|x) = \prod_{i=1}^{n} P(y_i|MB(y_i)), \tag{9}$$

where n is the number of label variables in G, and the CPD $P(y_i|MB(y_i))$ can be factorized over all the cliques which contain Y_i:

$$P(y_i|MB(y_i)) = \frac{1}{Z_i(x_i)} \prod_{Y_i \in c_j} \phi_j(y_i, v_j), \tag{10}$$

where v_j is the values of all variables in clique c_j except for Y_i, and Z_i is the local partition function (or normalization constant) given by

$$Z_i(x_i) = \sum_{y_i'} \prod_{Y_i \in c_j} \phi_j(y_i', v_j). \tag{11}$$

We see that computing pseudo-likelihood is very efficient because the local partition function is simply a sum over a single variable.

The log of the MAP objective function based on the pseudo-likelihood of the training data is as follows:

$$\begin{aligned} PL(\mathcal{I}, w) &= \log\Big(P(y|x)\prod_k P(w_k)\Big) \\ &= \log P(y|x) + \sum_k \frac{-w_k^2}{2\sigma^2} - \log\sqrt{2\pi\sigma^2} \\ &= \sum_{i=1}^{n}\Big(\sum_{Y_i \in c_j} w \times f(y_i, v_j) - \log Z_i\Big) - \\ &\quad \frac{\|w\|_2^2}{2\sigma^2} + C. \end{aligned} \tag{12}$$

Similar to MLE, gradient-based optimization algorithms are used to maximize the log-pseudo-likelihood. For computing the gradient of the objective function, the partial derivative of $PL(\mathcal{I}, w)$ with respect to w_k is given by

$$\begin{aligned} &\frac{\partial PL(\mathcal{I}, w)}{\partial w_k} \\ &= \sum_{i=1}^{n}\sum_{Y_i \in c_j}\Big\{f_k(y_i, v_j) - E_{P_w}\big[f_k(y_i, v_j)\big]\Big\} - \frac{w_k}{\sigma^2}, \end{aligned} \tag{13}$$

where the expected feature values is related to P_w:

$$E_{P_w}\big[f_k(y_i, v_j)\big] = \sum_{y_i'}\{f_k(y_i', v_j)P_w(y_i', v_j)\}. \tag{14}$$

Computing the gradient in (13) is simply a sum over each possible value of each variable and does not require running the inference procedure, which is much more efficient than computing the gradient of MLE objective function in (7).

3.2.4 Inference in CRFs

The task of inference in a CRF is to estimate the most likely assignment of the target variables given an input x and parameters w. This task can be solved by belief propagation (BP) algorithm[17]. The BP algorithm was originally proposed in the context of Bayesian networks. It calculates the marginal distribution for each label variable approximately by local message passing, and then the likelihood and its gradient can be computed based on the marginals. The BP algorithm generates provably correct results if the graph has no loops, while if the graph contains loops, then the algorithm (so called loopy BP) might not converge to the correct probability distribution[17]. But the empirical results[18] show that the loopy BP algorithm can converge to a good approximation to the correct marginals in most of the time.

For making the inference procedure converge more quickly, we use an iterative inference algorithm, which is very similar to the loopy BP algorithm but much briefer. We initialize the marginals and the values of label variables by using only the content attributes, and then update them iteratively with the state of the variables at the previous time, until all the label variables do not change any more. We use $Y_i^{(t)}$ to denote the state of Y_i at step t and $P_{(y_i)}^{(t)}$ to denote the probability of $Y_i^{(t)} = y_i$. The detailed procedure of the iterative inference algorithm is presented in Algorithm 1.

Algorithm 1. Iterative Inference Algorithm
Input: content attributes $x = \{x_{\text{call\_workday}\,t_n}, x_{\text{call\_holiday}\,t_n}, x_{\text{sm\_workday}\,t_n}, x_{\text{sm\_holiday}\,t_n}, x_{\text{sp\_workday}\,t_n}, x_{\text{sp\_workday}\,t_n}, x_{\text{sp\_holiday}\,t_n}\}$, $t_n \in T(N)$; parameters w;
Output: labels Y_i; marginals $P(y_i)$;
//initiation:
foreach label variable Y_i **do**
  foreach possible value y_i **do**
    //initialize the marginal by using only the content
    //attributes x_i
    $P^{(0)}(y_i) \leftarrow \prod_{x_{ik} \in x_i} \phi_{c_{ik}}(x_{ik}, y_i)/Z_i(x_i)$;
    //where $\phi_{c_{ik}}(x_{ik}, y_i) = \exp\{y_i w_k x_{ik}\}$ and $Z_i(x_i)$
    //$= \sum_{y_i'} \prod_{x_{ik} \in x_i} \phi_{c_{ik}}(x_{ik}, y_i')$
  end
  $Y_i^{(0)} \leftarrow \arg\max_{y_i'} P^{(0)}(y_i')$;
end
//iteration:
repeat
  foreach label variable Y_i **do**
    foreach possible value y_i **do**
      //update the marginal by using the state of

```
        //the variables at step t − 1 (i.e., v^(t−1) =
        // x^(t−1) ∪ y^(t−1) and let v_j^(t−1) be the state
        // of variables in clique c_j at step t − 1
        P^(t)(y_i) ← ∏_{c_j:Y_i∈c_j} φ_{c_j}(y_i, v_j^(t−1))/Z_i(v^(t−1));
        //where φ_{c_j}(y_i, v_j^(t−1)) = exp{w_k f_k(y_i, v_j^(t−1))}
        //(see (5) about f_k(y_i, v_j^(t−1)))and
        //Z_i(v^(t−1)) = Σ_{y'_i} ∏_{c_j:Y_i∈c_j} φ_{c_j}(y_i, v_j^(t−1))
    end
    Y_i^(t) ← arg max_{y'_i} P^(t)(y'_i);
  end
until each variable Y_i satisfies Y_i^(t) = Y_i^(t−1);
```

4 Typed Community Detection

In this section, we discuss discovering typed communities based on the results of relationship labeling. There are two methods.

4.1 Method 1: Relationship Cutting

This is a very simple method which directly uses the relationship labels. We retain the relationships with a particular label in a mobile social network but remove all the remaining relationships with other labels, and then the network is partitioned into many connected components and the typed communities are generated naturally. For example, if we retain the family relationships and remove other ones, then we get the family communities.

The disadvantage of this method is that the community detection accuracy is completely dependent on the accuracy of relationship labeling without considering the link structure of a mobile social network, and the mislabeling of some key relationships may lead to some serious errors in typed community detection.

4.2 Method 2: Weighted Community Detection

The basic idea of general community detection is using the link structure of a network to find a meaningful division of nodes (communities) which are more interconnected relative to their connectivity to the rest of the network. Many sophisticated community detection algorithms have been developed to tackle this problem in recent years, including modularity-based algorithms, Spectral algorithms, methods based on statistical inference, label propagation algorithm (LPA), etc. Fortunato[19] provides a detailed summary.

According to whether intersecting communities are generated, community detection algorithms can be divided into non-overlapping and overlapping algorithms. Non-overlapping community detection supposes that every node only belongs to a single community, while overlapping community detection allows a node to belong to more than one community. For the typed community detection problem, we can choose non-overlapping or overlapping community detection algorithms according to the types of communities we are discovering. In this paper, we just focus on discovering the family and corporate communities in mobile social networks. Since a mobile user belongs to only one family or corporation in general, we do not consider the situation of overlapping.

General community detection algorithms assume that all the edges in a network are equal, while the weighted ones take into account the different influence of edges with different weights to the community structure of the network. We know that the results of inference in a CRF are the marginals of the target variables. Based on the marginals we can compute the probability of each possible label for a relationship, and then the label with the maximum probability is identified as the label of the relationship. Such a probability indicates the tightness of a relationship in a certain type. Consequently, we can use these probabilities of labels as the weights of relationships:

$$w_i^{(y_k)} = P(Y_i = y_k).$$

Then we can get a weighted network for each possible label. Fig.7 illustrates a simple example of weighted social networks, in which the solid lines indicate the family relationships while the dashed lines indicate the colleague relationships. After that, some sophisticated

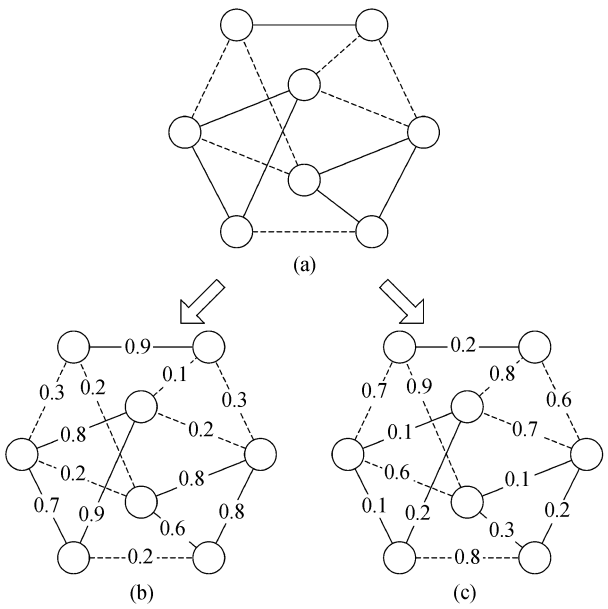

Fig.7. Simple example of weighted social networks. (a) Example of a social network. (b) "Family" social network. (c) "Colleague" social network.

weighted community detection algorithms can be employed on each weighted network and the typed communities are generated naturally.

Discovering the typed communities in mobile social networks by using weighted community detection algorithms based on the results of relationship labeling, which can amend some inaccuracies of relationship labeling by utilizing the link structure of the network, should be a very effective method. It is to be noted that this method requires the results of relationship labeling are represented in a form of probability.

5 Experiments

In this section, we present a set of experiments to evaluate our typed community detection framework. We performed the experiments on two real-world datasets.

5.1 Datasets

We collected two mobile social networks from the encrypted mobile user logs obtained from a mobile operator in China. The logs recorded all the communication and movement events of users occurred within about 3 weeks in July 2010. And all the users were from a small city in northern China.

For training and testing our method, the relationships in the datasets should be labeled in advance and the community structures should be known. Manually labeling the relationships in the datasets was not easy. It was lucky that we could do this by using the service packages (e.g., family packages or group packages) provided by the mobile operator. The first dataset we collected consisted of a number of user groups which ordered a family package plus the relationships between the users. We labeled a relationship with "family" if it was between two users in the same family package, otherwise it was labeled with "non-family". Similarly, the second dataset consisted of a number of user groups which ordered a group packages, and a relationship between two users in the same group package was labeled with "colleague" while a relationship between two users in different group packages was labeled with "non-colleague".

The summary of the datasets is shown in Table 1.

Table 1. Summary of Datasets

	Dataset 1	Dataset 2
Number of users	243	372
Number of relationships	421	1 923
Number of communities	72	33
Average community size	3.375	11.273
Number of inner-community relationships	296	1 384
Number of inter-community relationships	125	539

Note: The inner-community relationships in dataset 1 are family relationships while the inter-community relationships are non-family relationships. For dataset 2, the inner-community relationships are colleague relationships and the inter-community relationships are non-colleague relationships.

5.2 Experimental Setup

In relationship labeling, two traditional content-only (CO) classifiers, naïve Bayes and logistic regression (respectively denoted as CO-NB and CO-LG), are used as the baseline, which use only the contents of relationships and do not take the network into account. Furthermore, we compared the CRF model with the iterative classification algorithm (ICA)[20-21], which makes use of a local classifier for classifying an object by utilizing both its content and the labels of its neighbors. Also naïve Bayes and logistic regression are chosen as the local classifiers in the ICA algorithm (respectively denoted as ICA-NB and ICA-LG).

For the feature construction, we set the duration of period of time $d(N)$ (see Definition 1 in Section 2) to be 15 minutes, 30 minutes, 1 hour, 2 hours, 4 hours and 8 hours, then a day was divided into $N = 96$, 48, 24, 12, 6 and 3 periods of time respectively.

For each of the two experimental networks, we randomly split the relationships into two subnetworks. In the relationship labeling phase, we used one subnetwork for training and the other for test, and then swapped. We trained and tested a binary classifier using each classification model on each network.

In the typed community detection phase, we employed the Infomap algorithm[22] as the weighted community detection algorithm. This algorithm finds the best cluster structure of a graph by optimally compressing the information describing the probability flow of random walk and has a complexity that is essentially linear in the size of the graph. It is considered as one of the best community detection algorithms so far[23].

We used the best results of relationship labeling for typed community detection. And for maintaining the structural integrity, we remerged the two subsets of each dataset to be an intact network.

The last problem was the evaluation of the discovered communities. We got the real community structure of each network through the service packages, i.e., all the users who ordered the same package formed a community. In [24], the fraction of correctly classified vertices is used as a community detection evaluation. But this measure is too harsh, since only the largest community found within each of the real communities is considered correctly classified. Normalized mutual information (NMI)[25], a measure of similarity between the discovered and the real community structures based

on information theory, is employed to evaluate the performance of our proposed framework in this paper.

In addition, we propose a new measure, i.e., *fraction of correctly classified node pairs*, to calculate the similarity between the discovered and the real community structures. For each pair of nodes, if they were simultaneously in or not in the same discovered community and the same real community, we think the node pair was correctly classified, otherwise it was incorrectly classified. Then the measure can be written as:

$$A = \frac{\sum_i \sum_j s_{ij}}{n(n-1)},$$

where n is the number of nodes, and s_{ij} is a binary function to track if the node pair (i, j) are correctly classified:

$$s_{ij} = \begin{cases} 1, & \text{if node pair } (i,j) \text{ is correctly classified,} \\ 0, & \text{otherswise.} \end{cases}$$

If the discovered communities are identical to the real communities, then all the node pairs are correctly classified and A takes its maximum value of 1. If the discovered community structure is totally independent of the real community structure, i.e., all the node pairs are incorrectly classified, then A takes its minimum value of 0.

The proposed measure does not simply count if a node is classified into the correct community; instead, it takes into account the topological relation between any pair of nodes (i.e., if they are in the same community) and provides a way to calculate the similarity between the discovered and the real community structures from the perspective of local topology information. We believe that it is a simple and effective evaluation.

Each experiment in this study was repeated 10 times and the results were averaged.

5.3 Experimental Results

The relationship labeling accuracies of the various classifiers on the two datasets are shown in Figs. 8 and 9.

Overall, the prediction accuracy of relationship labeling on dataset 1 is better than that on dataset 2. This shows that the label "family" is more sensitive to the communication and movement patterns between mobile users.

With the increase of the duration of period of time $d(N)$, the accuracies of all the classifiers descend, and at the same time the differences between the classifiers narrow gradually. That is because the features used in relationship labeling are constructed based on the communication and movement information of users in each period of time. The longer the duration of period of time is, the less effective the features are.

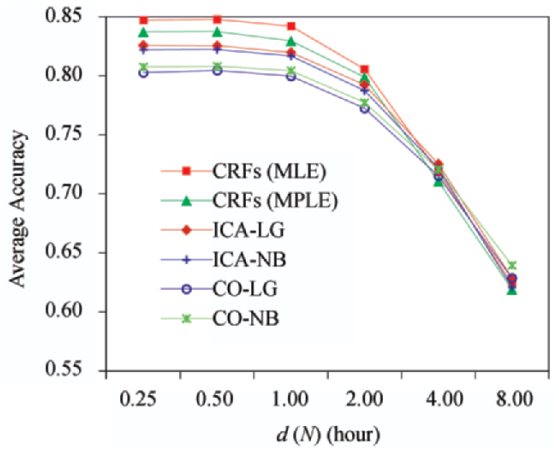

Fig.8. Average relationship labeling accuracy on dataset 1.

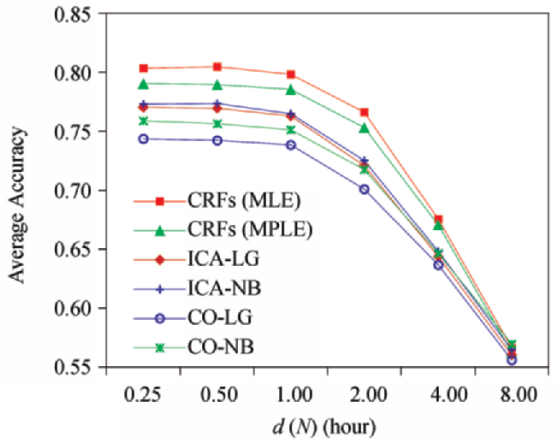

Fig.9. Average relationship labeling accuracy on dataset 2.

When appropriate duration of time is chosen, i.e., $d(N) \leqslant 2$ hours, the CRF model outperforms the other methods (increases the prediction accuracy by around 4∼6% over the content-only methods and 3∼4% over the ICA algorithms on the two datasets). This demonstrates that the CRF model is a very effective tool for relationship labeling in mobile social networks.

The performance of the CRF model learnt by MPLE is a little worse than that of the model learnt by MLE. However, the training time of MPLE is much less than that of MLE. For instance, when $d(N) = 0.5$ hours, the average training times of MPLE on the two datasets are about 3.5 and 8.0 seconds respectively, while those of MLE are about 48.0 and 160.0 seconds (all the results were computed on a PC with CPU 3.0 GHz and 2 GB RAM).

The typed community detection accuracies of the various approaches on the two datasets are shown in

Figs. 10 and 11. The non-weighted Infomap algorithm is used as a baseline, which does not take the labels of relationships into account and has nothing to do with the duration of period of time. Consequently, it cannot identify the type of a discovered community.

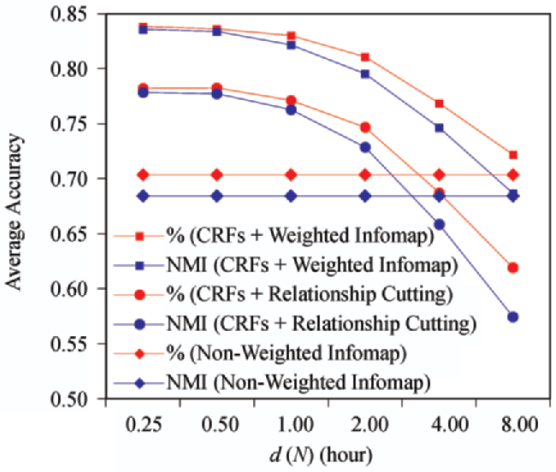

Fig.10. Average accuracy of typed community detection on dataset 1. Here % denotes the fraction of correctly classified node pairs.

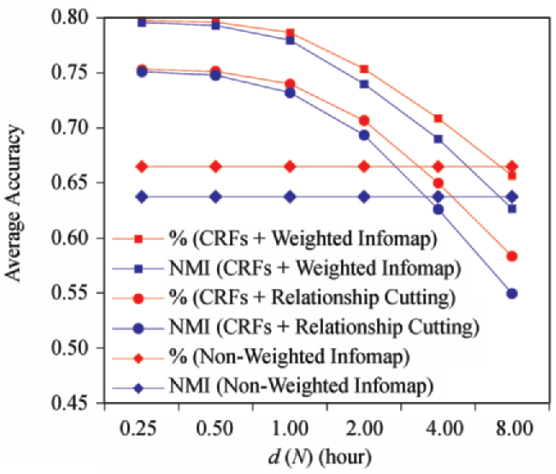

Fig.11. Average accuracy of typed community detection on dataset 2.

No matter which approach is used, the accuracy of family community detection is better than those of corporate community detection overall. That may be because the average tightness of corporate community is less than that of family community. Another reason for the relationship labeling based approaches is that the labeling accuracy on dataset 1 is better than that on dataset 2.

From Figs. 10 and 11 we can see that the weighted community detection based on relationship labeling is a very effective approach to discovering typed communities in mobile social networks, especially when the accuracy of relationship labeling is good. Directly cutting relationships based on the results of relationship labeling is also an acceptable manner when the labeling accuracy is good enough, but it is very bad when the labeling accuracy is very low.

6 Conclusions

In the paper we proposed discovering typed communities in mobile social networks based on the labels of relationships between users. We used user logs stored by mobile operators, including communication and user moving track information, to collectively label all the relationships in a network, by employing an undirected probabilistic graphical model, i.e., conditional random fields. Then we used two methods to discover typed communities based on the results of relationship labeling. One is simply retaining or cutting relationships according to their labels and the other is using sophisticated weighted community detection algorithms.

The experimental results on two real-world datasets show that our proposed framework performs well in terms of the accuracy of typed community detection in mobile social networks.

Of course, our proposed framework can be applied in other network fields such as online social networks. For example, we can identify the types of relationships between users in online social networks according to their ages, genders, educations, specialties, interests, behaviors, blogs, etc., and then discover the communities with different types.

References

[1] Wasserman S, Faust K. Social Network Analysis. Cambridge, UK: Cambridge University Press, 1994.

[2] Scott J. Social Network Analysis: A Handbook (2nd edition). London, UK: Sage Publications, 2000.

[3] Dong Z, Song G, Xie K, Wang J. An experimental study of large-scale mobile social network. In *Proc. the 18th International Conference on World Wide Web* (*WWW 2009*), Madrid, Spain, April 20-24, 2009, pp.1175-1176.

[4] Wang Y, Cong G, Song G, Xie K. Community-based greedy algorithm for mining top-K influential nodes in mobile social networks. In *Proc. the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD 2010*), Washington, DC, USA, July 25-28, 2010, pp.1039-1048.

[5] Zheng V W, Zheng Y, Xie X, Yang Q. Collaborative location and activity recommendations with GPS history data. In *Proc. the 19th International Conference on World Wide Web* (*WWW 2010*), Raleigh, NC, USA, April 26-30, 2010, pp.1029-1038.

[6] Ministry of Industry and Information Technology of the People's Republic of China. http://www.miit.gov.cn/.

[7] Girvan M, Newman M E J. Community structure in social and biological networks. *PNAS*, 2002, 99(12): 7821-7826.

[8] Newman M E J. Detecting community structure in networks. *The European Physical Journal B*, 2004, 38(2): 321-330.
[9] Hui P, Xu K, Li V O K, Crowcroft J, Latora V, Lio P. Selfishness, altruism and message spreading in mobile social networks. In *Proc. IEEE INFOCOM Workshops 2009*, Rio de Janeiro, Brazil, April 19-25, 2009, pp.1-6.
[10] Lafferty J, McCallum A, Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. the 18th International Conference on Machine Learning* (*ICML 2001*), Williams College, Williamstown, MA, USA, June 28-July 1, 2001, pp.282-289.
[11] Jensen D, Neville J, Gallagher B. Why collective inference improves relational classification. In *Proc. the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD 2004*), Seattle, WA, USA, Aug. 22-25, 2004, pp.593-598.
[12] Macskassy S A, Provost F. Classification in networked data: A toolkit and a univariate case study. *The Journal of Machine Learning Research*, 2007, 8: 935-983.
[13] Ng A Y, Jordan M I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14* (*NIPS 2001*), Vancouver, British Columbia, Canada, Dec. 3-8, 2001, pp.841-848.
[14] Lu Q, Getoor L. Link-based classification. In *Proc. the 20th International Conference on Machine Learning* (*ICML 2003*), Washington, DC, USA, Aug. 21-24, 2003, pp.496-503.
[15] Besag J. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society, Series D* (*The Statistician*), 1975, 24(3): 179-195.
[16] Getoor L, Taskar B. Introduction to Statistical Relational Learning. Cambridge, USA: The MIT Press, 2007.
[17] Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Francisco, USA: Morgan Kaufmann, 1988.
[18] Murphy K P, Weiss Y, Jordan M I. Loopy belief propagation for approximate inference: An empirical study. In *Proc. the 15th Conference on Uncertainty in Artificial Intelligence* (*UAI 1999*), Stockholm, Sweden, July 30-Aug. 1, 1999, pp.467-475.
[19] Fortunato S. Community detection in graphs. *Physics Reports*, 2010, 486(3-5): 75-174.
[20] Neville J, Jensen D. Iterative classification in relational data. In *Proc. the AAAI-2000 Workshop on Learning Statistical Models from Relational Data* (*SRL 2000*), Austin, TX, USA, July 30-Aug. 3, 2000, pp.13-20.
[21] Sen P, Getoor L. Link-based classification. Technical Report CS-TR-4858, Department of Computer Science, University of Maryland, 2007.
[22] Rosvall M, Bergstrom C T. Maps of random walks on complex networks reveal community structure. *PNAS*, 2008, 105(4): 1118-1123.
[23] Lancichinetti A, Fortunato S. Community detection algorithms: A comparative analysis. *Physical Review E*, 2009, 80(5): 056117.
[24] Newman M E J, Girvan M. Finding and evaluating community structure in networks. *Physical Review E*, 2004, 69(2): 026113.
[25] Danon L, Díaz-Guilera A, Duch J, Arenas A. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005: P09008.

Huai-Yu Wan received his B.S. and M.S. degrees in computer science from Beijing Jiaotong University in 2004 and 2007 respectively. He is currently a Ph.D. candidate in computer science and technology at Beijing Jiaotong University. His research interests focus on data mining and social network analysis.

You-Fang Lin received his Ph.D. degree in computer science and technology from Beijing Jiaotong University in 2003. He is currently an associate professor, vice dean of the School of Computer and Information Technology, Beijing Jiaotong University. His research interests cover data warehousing, data mining, intelligent system, complex network and social network analysis.

Zhi-Hao Wu received the B.Sc. degree in computer science and technology from Beijing Jiaotong University in 2007. Now he is a Ph.D. candidate in computer science and technology at Beijing Jiaotong University. His current research focuses on community detection and evolution in complex networks.

Hou-Kuan Huang was born in 1940. He is a professor in the School of Computer and Information Technology at Beijing Jiaotong University. He is a senior member of CCF. His research interests include artifical intelligence, pattern recognition, data mining and multi-agent system.