IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 14, 2019, accepted October 7, 2019, date of publication October 17, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948073

Factor Graph Model Based User Profile Matching Across Social Networks

LIDONG WANG[1], KEYONG HU[1], YUN ZHANG[2], AND SHIHUA CAO[1,3], (Member, IEEE)

[1]Qianjiang College, Hangzhou Normal University, Hangzhou 310018, China
[2]Institute of Zhejiang Radio and TV Technology, Zhejiang University of Media and Communications, Hangzhou 310018, China
[3]Faculty of Information and Technology, Macau University of Science and Technology, Taipa 000000, China

Corresponding author: Lidong Wang (violet_wld@163.com)

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY19F020022 and Grant LY17E070004, in part by the National Natural Science Foundation of China under Grant 61602402, in part by the Hangzhou Social Development Research Project under Grant 20170533B05, and in part by the Zhejiang Provincial Basic Public Welfare Research under Grant LGG19F020001.

ABSTRACT In modern society, it is common for people to be active in many different online social networks at once. As new social network services arise every year, it remains a great challenge to integrate social data. Discovering multiple profiles of a single person across different social networks is a precondition for integration, but it is still challenging due to the inconsistency and disruption of the accessible information among social media networks (SMNs). Many studies have made efforts on user's profiles, users' contents, and network structure to address this issue, but the issue of how to consider all these information in a unified model and tackle them simultaneously still remains challenging. Considering that identical users tend to have partial similar friend relationship structures in different SMNs, especially friendship SMNs, we deepen the analysis of ''friend'' relationships (mutual following connections) in different SMNs, and propose PIFGM (Pairwise Identical Factor Graph Model), a novel factor graph model-based model, to address this problem by considering both user attributes and friend relationships across networks. We also present a distributed learning algorithm to handle large-scale social networks. We evaluate the proposed model on two different data collections: SNS and SR. Our experimental results validate the effectiveness and efficiency of the proposed model. The proposed PIFGM significantly outperforms several alternative methods by up to approximately 10%∼20% in terms of F1 and precision on SNS and SR respectively.

INDEX TERMS Cross-platform, user profile matching, factor graph model, friend relationship, PIFGM method.

I. INTRODUCTION

In recent years, users have been introduced to many online social networks such as Sina microblog, Twitter, Instagram, or LinkedIn. Due to this diversity of online social media networks (SMNs), more and more people tend to use different SMNs for different purposes. People may be attracted by different functions offered by different social networks to better take advantage of services provided by each social network. For instance, users may use Twitter to publish opinions on political events while adopt Instagram to share their leisure activities. A survey in the US shows that two-thirds of online adults use social media platforms, such as Facebook, Twitter, MySpace, or LinkedIn, to stay in touch with friends, family members, and business partners. Users produce various content and also build different ego-networks in different networks.

With the rise of different online social networks being used for different purposes with different contexts, analyzing a user identity on a single social media may not give a comprehensive understanding of his/her personalities and interests. If we can link a user's identities from multiple networks, collect and analyze his/her information on these social networks together, we may have a more comprehensive view about the user and provide better recommendations and services. The primary topic for cross-platform SMN research is user identification across different networks. However, the current online social network data is big, noisy, incomplete and highly unstructured, and these social networks are independent of each other.

The associate editor coordinating the review of this manuscript and approving it for publication was Shirui Pan.

Thus, it is not a trivial task to link users across multiple social networks.

Although each social network structure is constructed for the user's specific objective and only reflects a part of his/her real world social circle, most users may have the same or similar attributes on different social networks, such as username, biography, emails, location, gender, profile photo and links to other web pages. No solution can identify all identical users. Some attributes may be used to identify a portion of users across multiple SMNs. In the early stage, researchers used the email address to find identical users in different SMNs [11]. Although email addresses are an effective attribute for the task, they are difficult to get due to privacy protection concerns. To solve this problem, more research is provided on accessible attributes, e.g., the user profile attributes [1]–[3], the user activities [4], [5]. All these works can identify a portion of identical users.

However, the accessible attributes among different SMNs have become increasingly inconsistent, incomplete and disruptive. Moreover, profile features may be easily impersonated by other users. Network structure is much more accessible than profile information [6]–[8]. Network structure features refer to the social network interactions between different users in the same online social network. Most current research focuses on local network, whose structure features can be built through the one-hop neighbors, such as following, followee and friend relationships. The single following relationships can not represent true "friend" relationships since a lot of users tend to follow famous persons in different SMNs. Obviously, friend relationships represents mutual-following connections [7], where each connection requires mutual confirmations between the two users, are much more reliable and consistent among microblogging SMNs, such as Sina and Tencent Microblog, Twitter, Facebook, Instagram, etc. Goga et al. [16] noted that many users have accounts on several SMNs (such as MySpace, Twitter, Facebook and Flickr etc.). We investigated 213 users with both Twitter and MySpace accounts and found that an average of 70.2% of their friends concurred in Twitter concurred in MySpace. Also, we conducted a similar investigation on 152 users with both Facebook and Flickr accounts, and found an average of 61.3% overlapping on their friend circle. Since identical users tend to set up partial similar friend relationship structures in different SMNs, it may be much more suitable for cross-platform user identification tasks.

The joint use of profile information and network structure may lead to better results [9], [10], [12]. Specifically, Bartunov et al. [12] integrated profile information with a network structure using a Conditional Random Fields model, but it only considered local identity resolution, that is, discovering matching profile pairs across the contacts of the given seed user, and it is difficult to scale for big data. Zhang et al. [10] proposed an energy based model to formalize the problem as a unified framework, it mainly focuses on the global consistency on user identification tasks among multiple SMNs. In this paper, we hope to formulate local accessible attributes and friend relationship matching into a novel principled optimization model. To the best of our knowledge, no previous work has explored the preceding problems to such extent.

However, the task remains challenging because of the unbalanced nature of users' information. First, the accessible attributes among different SMNs become increasingly inconsistent, partial and noisy. It is not ideal to identify users across networks only by attributes matching; Second, as real networks become larger and larger with millions of nodes, it is important to develop efficient algorithms that can scale up well. Third, the existing friend relationship definition only considers the neighborhood relationship between two users, it neglects the relationships among three or more users [8]. To address these challenges, in this paper, we conduct a systematic investigation into the problem of user identification across multiple social networks. We formally define the problem and develop a novel factor graph model to support our task. Our contributions can be summarized as follows:

(1) We deepen the analysis on friend relationships, and give the formal definition of dyadic friend relationship and triadic friend relationship. Based on this, we propose PIFGM (Pairwise Identical Factor Graph Model), a novel factor graph model, to formalize our problem as a unified optimization framework by considering both local profile attributes and friend relationships.
(2) We present a distributed implementation of the learning algorithm based on MPI (Message-Passing Interface) without information lose to scale up to large networks.
(3) Providing concrete demonstrations of PIFGM performance with two datasets: SNS network dataset [10] and SR network dataset. SNS consists of several popular social networks including Twitter, Flickr, Myspace, Last.fm, and Live-Journal. SR data collection consists of Sina Microblog and Renren. Findings show that PIFGM is superior to other representative methods in these datasets.

This article proceeds as follows. Section 2 systematically presents terminology on user identification across different SMNs. Section 3 reviews related works on cross-platform user identification. Section 4 proposes the PIFGM algorithm. Section 5 covers the experimental studies. Section 6 offers conclusions.

II. PROBLEM DEFINITION IN CROSS-PLATFORM USER IDENTIFICATION

A. PROBLEM DEFINITION

Let $SMN = (U, E, \mathbf{R})$ denotes a social network, where $U = \{u_1, u_2, \ldots, u_N\}$ is a set of N users and E is a set of relationships between users. Each element $e_{ij} = (u_i, u_j) \in E$ represents a mutual following relationship between the user u_i and the user u_j. Each user is associated with a $d-$dimensional attribute vector $\mathbf{r_i}$ (the i-th row in $\mathbf{R}$), which is defined based on the user's profile information, such as names or emails. As more than one SMNs are discussed in this paper, $SMN^A = (U^A, E^A, \mathbf{R}^A)$ is used to represent SMN A. Throughout this

TABLE 1. Notations.

Symbol	Description
$SMN = (U, E, \mathbf{R})$	Social network with user set U, relationship set E and attribute matrix $\mathbf{R}$.
SMN	A set of social networks.
U	A set of $\|U\| = N$ users.
E	A set of relationships between users.
$\mathbf{R}$	A $N \times d$ attribute matrix.
$X = \{x_i\}$	A set of candidate user pairs.
$Y = \{y_i\}$	A set of corresponding binary label for X.
Y^L	A set of labeled identical user pairs.
u_i^A	User i in SMN A.

paper, we use the superscript ^{A} to indicate variables (or notations) associated with the *SMNA*. The notations frequently used in this paper are listed in Table 1. Moreover, the set items and the vectors are marked in italic, while the matrices are presented in bold. Given this, we define the task of user identification as follows: Given two social networks $\mathbf{SMN} = \{SMN^A, SMN^B\}$, a user $u_i^A \in U^A$ and a user $u_j^B \in U^B$, decide whether u_i^A and u_j^B refer to the same user. An example of such task is shown in Figure 1. White node pairs are labeled pairs, gray nodes are not projected, blue node pairs are discovered by our method.

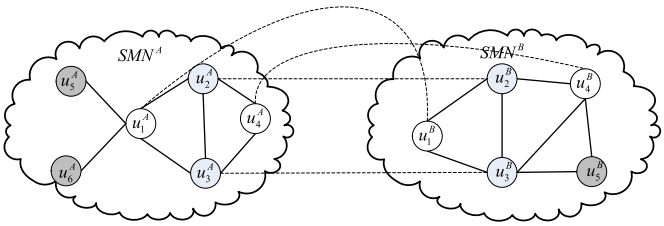


FIGURE 1. An example of user identification across two networks.

For convenience, we have the following definitions.

Definition 1 (Candidate User Pair): u_k^A and u_j^B from two different social networks, SMN ^{A} and SMN ^{B}, form a candidate user pair. We denote $X = \{x_i\}$ as the set of all candidate pairs from SMN ^{A} and SMN ^{B}, where $x_i = (u_k^A, u_j^B)$.

Definition 2 (Identical Users): For any $x_i \in X$, if two users in the user pair x_i refer to the same individual in real life, then they are identical users. Let $Y = \{y_i\}$ be the set of corresponding binary labels in $\{0, 1\}$. $y_i = 1$ means the the user pair x_i refer to the same user, otherwise not.

Given two social networks $\mathbf{SMN} = \{SMN^A, SMN^B\}$, the objection of user identification is to learn a function

$$f : (\mathbf{SMN} = \{SMN^A, SMN^B\}, X, Y^L) \rightarrow Y \qquad (1)$$

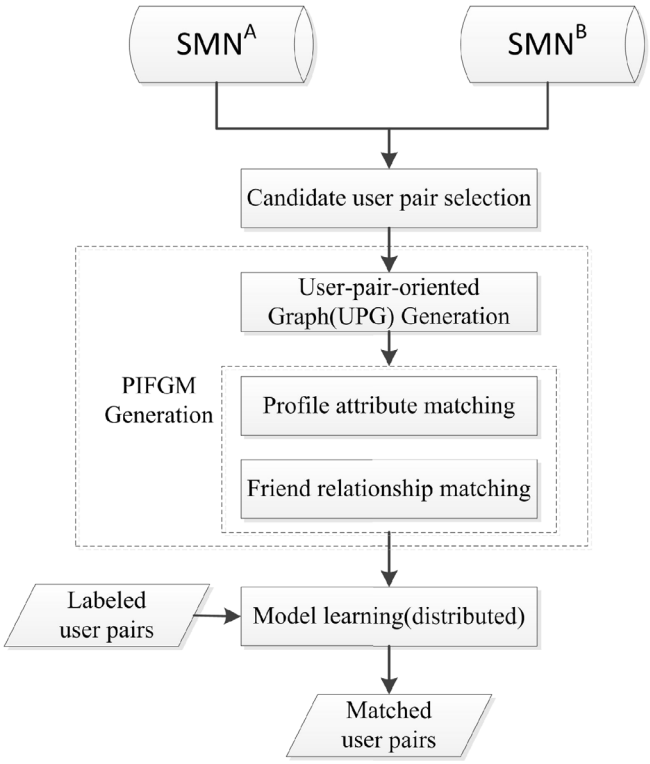


FIGURE 2. Framework of PIFGM.

that can be used to determine y_i for each candidate user pair. For simplicity, in this work, we assume that one user has at most one account in an online social network.

B. THE FRAMEWORK OF PIFGM

According to the definitions above, identifying users across two SMNs yields a set of labeled identical user pairs and candidate user pairs. Figure 2 shows the framework of our proposed method.

It contains three modules: candidate user pair generation, Pairwise Identical Factor Graph Model (PIFGM) generation, and model learning for PIFGM. We select a set of candidate user pairs based on attributes and structure information to reduce the size of UPG. The PIFGM generation contains three steps: UPG generation, profile attribute matching and friend relationship matching. The objective of PIFGM generation is to model the correlations between two or three candidate user pairs. We factorize the joint distribution over the label set in UPG into a product of factors, which are attribute factor function and friend relationship factor function. During the model learning, we maximize the log-likelihood function of all the labeled user pairs, and use gradient descent algorithm to learn the parameters. Furthermore, we develop a distributed learning method based on MPI (Message Passing Interface) to scale up well with large networks.

III. RELATED WORKS

We reviewed current studies on cross-platform user identification from three categories: profile-based approaches, structure-based approaches and hybrid approaches.

A. PROFILE-BASED APPROACHES

The use of the attributes to match users across distinct social networks has been largely investigated [13]–[17]. Some researchers focus only on the username of an individual as a way to match different profiles [13], [14]. Perito et al. [13] explored the possibility of linking users profiles only by looking at their usernames. Zafarani [14] developed a user mapping method by modeling user behavior on screen names. Despite that the identification task depends on username analysis is easy to operate, some researchers found that Leveraging a combination of profile features can result in better results. Motoyama et al. [15] described each pair of profiles as a vector of scores, which represent the similarity between the values of the attributes (e.g., email, gender, age, hometown). Goga et al. [16] showed that usernames, real names, locations, and photos can robustly identify about 80% of the matching pairs of user accounts between two social networks. Abel et al. [17] aggregated user profiles and matched users across systems. Raad et al. [21] matched user profiles by considering all the profile's attributes. Mu et al. [3] used the user's attributes (e.g., gender, birthday, and educational background) to explore a new concept of "Latent User Spac" to more naturally model the relationship between the underlying real users. Vozecky et al. [19] represented user profiles from Facebook and StudiVZ as n-dimensional vectors. Then, the vectors are compared by means of exact matching, partial matching and fuzzy matching. Wang et al. [24] used real name, username, location, and URL to generate the feature of the user, and applied several similarity measures to measure the similarity of the user profiles on different websites.

Besides, some researchers use writing style and a few semantic features to identify users [20]. Riederer et al. [5] presented an algorithm which utilizes trajectory-based features to link identical users. Li et al. [18] addressed the problem of user identification based on User Generated Content (UGC). Soroush et al. [22] was inspired by stylometry techniques and presented the models for Digital Stylometry to match user accounts. Such content-based features are often sparse in SMNs. Although public profile attributes provide powerful information for user identification, some attributes are difficult to obtain because of privacy protection. Thus, it is difficult to define attribute features to provide sufficient generality for different data collections.

B. STRUCTURE-BASED APPROACHES

Structure features are more easily to obtain than attribute features in social networks. To analyze privacy and anonymity, Narayanan et al. [23] developed NS, based solely on network topology. Zhou et al. [7] proved that using a friendship structure to analyze cross-platform SMNs is more effective than existing structure-based user identification scheme, but this method needs extra prior knowledge, and the performance on real-world networks is not satisfactory. In the following research, Zhou et al. [8] proposed an unsupervised scheme, named friend relationship-based user Identification algorithm without prior knowledge. In terms of distributed computing, Korula et al. [25] designed a local distributed algorithm that uses only structural information about the graphs to expand the initial set of links into a mapping/identification of a large fraction of the nodes in the two networks.

C. HYBRID APPROACHES

To improve existing attribute-based UIR methods, some researchers employed additional data sources, in particular social linkage data. Bennacer et al. [11] presented an algorithm that uses the network topology and the publicly available personal information to iteratively match profiles across social networks. Bartunov et al. [12] proposed a new approach for user profile matching based on Conditional Random Fields that extensively combines usage of profile attributes and social linkage. Zhang et al. [10] presented an energy-based model to link user identities by considering local features, network structure features and global consistency. Liu et al. [27] built structure consistency models to maximize the structure and behavior consistency on users' core social structure across different platforms. Most of the above methods need prior knowledge, such as labeled pairs or seed pairs. Fu et al. [28] proposed a graph node similarity measurement in consideration with both graph structure and descriptive information. This method does not need prior knowledge, but the time complexity is pretty high. Malhotra et al. [26] combined the user profile information and friend network to generate the user's digital footprints, and applied automated classifiers for user identification based on user's footprints.

Except for the above research, some researchers deanonymize social network users by graph matching algorithms [29], [30]. However, these methods mostly focus on the synthetic graphs. In this study, we propose an innovative hybrid approach PIFGM by considering both local profile attributes and network structures, which are similar to JLA and COSNET. However, our method differs from these two methods in the following aspects:

(1) JLA and PIFGM are suitable for finding identical users across two different networks, while COSNET mainly focuses on the global consistency among more than three networks.
(2) JLA utilizes Conditional Random Field that combines usage of profile attributes and social linkage, COSNET utilizes energy-based model by considering both local and global consistency among multiple networks, while PIFGM utilizes factor graph model to formalize the task as a as a unified optimization framework.
(3) Although three algorithms utilize both local attributes and network structure, the detailed features are different. The local attributes mostly depend on the available information on certain networks. In other words, different networks may show different attribute features. In terms of network structure, JLA depends on the number of sharing friends, which may probably mismatch users when

they share few known friends. COSNET only considers the neighborhood relationship between two users, and neglects the relationship among three users. PIFGM converts the connections in different SMNs into two kinds of friend relationships in User Pair Graph (UPG), defined as dyadic friend relationship and triadic friend relationship, are more effective for user identification tasks.

(4) PIFGM and COSNET can be easily implemented in a distributed learning manner, while JLA cannot. We propose a distributed learning method based on MPI without information loss.

IV. PAIRWISE IDENTICAL FACTOR GRAPH MODEL

In general, there are two ways to model the problem. The first way is to model each user from two SMNs as a node. For each node we try to estimate probability distributions of the relationship (whether they are the same person) from the user in SMN^A to another user in SMN^B. The graphical model consists of $M+N$ variable nodes, where M, N denote the number of users in SMN^A and SMN^B respectively. This model is intuitive, but it suffers from some limitations. For example, it is difficult to model the neighborhood relationships between two pairs of identical users, and its computational complexity is high. An alternative way is to model each candidate user pair from two SMNs as a node in a graphical model, which is used in our paper. Based on this, we propose a Pairwise Identical Factor Graph Model (PIFGM) by considering both attribute matching and friend relationship matching.

The basic idea of PIFGM is to incorporate local profile features and network structures as factors in a factor graphical model. We transform the original node-oriented network into a user-pair-oriented graph (UPG=($\mathrm{U_{UPG}}$, $\mathrm{E_{UPG}}$)) by formalizing each candidate user pair as a node, where $x_i \in U_{UPG}$, $e_{ij} = (x_i, x_j) \in E_{UPG}$. Let $\mathrm{x}_1 = (u_i^A, u_j^B) \in U_{UPG}$ and $\mathrm{x}_2 = (u_k^A, u_l^B) \in U_{UPG}$ be two candidate user pairs (two nodes) from SMN^A and SMN^B, we add an edge between these two nodes if

$$u_i^A \in \Gamma(u_k^A), \quad u_j^B \in \Gamma(u_l^B)$$

where $\Gamma(\cdot)$ represents the neighborhood set. The PIFGM is constructed based on the UPG. Figure 3 gives an example of a UPG generated from two networks and the constructed PIFGM model. The relationships between x and its corresponding user pair are listed in Table 2. As shown in Figure 3(a), both two input networks SMN^A and SMN^B contain three users, then 9 candidate user pairs can be generated. Figure 3(b) represents a generated UPG based on the 9 candidate user pairs.

Thus, we construct a factor graph in Figure 3(c) to model the correlations between two or three candidate user pairs[1]. The model is referred to as a Pairwise Identical Factor Graph Model. Given a UPG, we factorize the joint distribution over

[1]A factor graph is a particular type of graphical model, which takes a bipartite graph to represent the factorization of a joint distribution and enables efficient computation of marginal distributions through the sum-product algorithm.

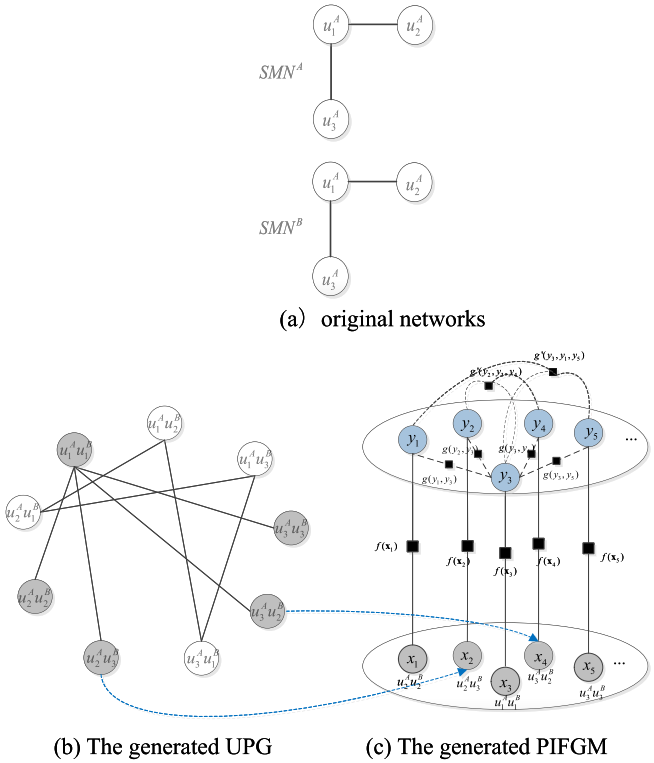


FIGURE 3. Illustration of the generation of UPG and the generation of PIFGM.

TABLE 2. Details of x.

x	The corresponding user pair
x_1	(u_2^A, u_2^B)
x_2	(u_2^A, u_3^B)
x_3	(u_1^A, u_1^B)
x_4	(u_3^A, u_2^B)
x_5	(u_3^A, u_3^B)
x_6	(u_2^A, u_1^B)
x_7	(u_1^A, u_2^B)
x_8	(u_1^A, u_3^B)
x_9	(u_3^A, u_1^B)

the label set Y in UPG into a product of factors, with each factor representing a function of a set of variables, where a variable indicates the candidate user pair or the label of a user pair in the graph:

$$p(Y|UPG) = \frac{1}{Z}\prod_i f(\mathrm{x}_i, y_i)\prod_{\parallel i,j} g(y_i, y_j)\prod_{\Delta_{i,j,k}} g'(y_i, y_j, y_k) \tag{2}$$

where $f(\mathrm{x}_i, y_i)$ is the factor function representing the correlation between the candidate user pair x_i and the corresponding label y_i. Notation $g(y_i, y_j)$ denotes the factor function

corresponding to two candidate user pairs, which represents the correlation between two candidate user pairs. Notation $g'(y_i, y_j, y_k)$ denotes the factor function corresponding to three candidate user pairs in a triadic structure, which represents the correlation among three candidate user pairs. Notation Z denotes the global normalization term that adds up the products of the factor functions over all possible configurations of all the candidate user pairs' labels, i.e.,

$$Z = \sum_Y \prod_i f(\mathrm{x}_i, y_i) \prod_{\|_{i,j}} g(y_i, y_j) \prod_{\Delta_{i,j,k}} g'(y_i, y_j, y_k) \quad (3)$$

In the following, we introduce the details of the factor functions.

A. PROFILE ATTRIBUTE MATCHING

The attribute factor function $f(\mathrm{x}_i, y_i)$ is defined as follows:

$$f(\mathrm{x}_i, y_i) = \exp\{\alpha^{\mathrm{T}} \varphi(\mathrm{x}_i, y_i)\} \quad (4)$$

where $\varphi(\mathrm{x}_i, y_i)$ is a vector of feature functions for encoding the user profile similarity for the user pair x_i. The elements in $\varphi(\mathrm{x}_i, y_i)$ represent different attribute features. The parameter α is a D-dimensional weighting vector. The detailed features used in our paper are listed as follows. These features allow our model to be easily adapted to different data collections.

1) USERNAME SIMILARITY

The username is always publicly accessible, as it is a useful way to identify an individual within a social network, and is generally a part of the URL of the web page that hosts the profile. In order to determine the similarity of two usernames, we choose Levenshtein distance to measure username similarity, which is quite effective in capturing the variations in the usernames chosen by the individuals [13]. Then, the similarity of two usernames n_1 and n_2 is calculated as follows:

$$Sim(n_1, n_2) = 1 - \frac{lev(n_1, n_2)}{\max(l(n_1), l(n_2))} \quad (5)$$

where $lev(n_1, n_2)$ denotes Levenshtein distance, $l(n_1)$ is the number of characters of n_1.

2) USERNAME UNIQUENESS

Username uniqueness, denoted as $I(n) = -\log_2(p(n))$, which can quantify the amount of identifying information each username carries [13], where n denotes username. To calculate this, we apply Markov Chain techniques to calculate the probability of username $p(n)$. Assume that $n =' c_1 c_2 \ldots c'_m$, the probability of username n can be expressed as:

$$p(c_1, \ldots, c_m) = \prod_{i=1}^{m} p(c_i | c_{i-k+1}, \ldots, c_{i-1}) \quad (6)$$

It has to be noted that only the previous k characters are considered to compute the probability of the next character.

3) PROFILE CONTENT SIMILARITY

We can combine all the information related to a profile into a bag-of-words model.These information contain gender, birthday, location, educational background, etc. The similarity of two profiles is then measured by cosine distance.

B. FRIEND RELATIONSHIP MATCHING

Considering that friend relationship is robust, reliable and consistent in microblogging SMNs, we construct structure matching based on it. The basic idea of friend relationship matching is that if user u_i^A is matched onto u_j^B, then we hope that user u_i^A's friends in SMN^A can also be matched to u_j^B's friends in SMN^B. In graph theory, the problem can be reduced to graph isomorphism [10]. The objective of graph isomorphism is to find a structure-preserving bijection between the vertex sets of two graphs. However, the above idea does not consider the friend relationship among three users. According to the social balance theory [32], friends of friends tend to be the friends themselves, which composes a triadic structure in social networks. The triadic structure is the most basic group structure in social networks. A triad is a group of three people [33]. Roughly speaking, there are two types of triads: closed triads and open triads (see Figure 4). In a closed triad, there is a friend relationship between any two users. In an open triad, there are two relationships. If three users in SMN^A compose a triadic structure, and two of them in SMN^A are matched to two users in SMN^B, then we hope that the third ones in two networks are identical users. Thus, we introduce triadic structure to model the friend relationship among three users.

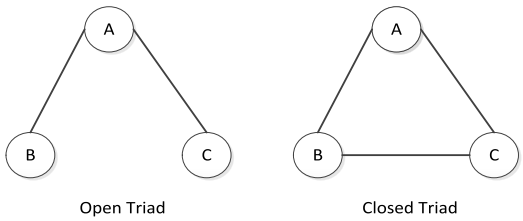


FIGURE 4. Open triad and closed triad. A, B and C represent users.

To leverage the friend relationships, we define the dyadic friend relationship and the triadic friend relationship.

Definition 3 (Dyadic Friend Relationship): Let $\mathrm{x}_i = (u_l^A, u_k^B)$, $\mathrm{x}_j = (u_m^A, u_n^B)$ be two nodes in a UPG, if $u_l^A \in N(u_m^A)$, $u_k^B \in N(u_n^B)$, which means that there is a connection between x_i and x_j, then we say that x_i and x_j have dyadic friend relationship. We denote this relationship as $\|_{i,j}$.

Definition 4 (Triadic Friend Relationship): Let $\mathrm{x}_i, \mathrm{x}_j, \mathrm{x}_k$ be three nodes in a UPG, if x_i, x_j and x_k compose a triad, then we say that x_i, x_j and x_k have triadic friend relationship. We denote this relationship as $\Delta_{i,j,k}$.

As shown in Figure 3(c), x_1 and x_3 have dyadic friend relationship, x_3, x_4 and x_2 have triadic friend relationship. Based on the above definitions, the factor function $g(y_i, y_j)$ and the factor function $g'(y_i, y_j, y_k)$ are respectively defined as follows:

$$g(y_i, y_j) = \exp\{\beta^{\mathrm{T}} \psi(y_i, y_j)\} \quad (7)$$

$$g'(y_i, y_j, y_k) = \exp\{\gamma^{\mathrm{T}} \xi(y_i, y_j, y_k)\} \quad (8)$$

where β and γ are the weighting vectors, $\psi(y_i, y_j)$ and $\xi(y_i, y_j, y_k)$ are one-hot vectors of dyadic friend relationship

and triadic friend relationship respectively, which are defined as follows:

$$\psi = (\psi_{0,0}, \psi_{0,1}, \psi_{1,0}, \psi_{1,1}) \tag{9}$$

$$\xi = (\xi_{0,0,0}, \xi_{0,0,1}, \xi_{0,1,0}, \xi_{0,1,1}, \ldots) \tag{10}$$

$$\psi_{m,n}(y_i, y_j) = \begin{cases} 1 & y_i = m,\ y_j = n \\ 0 & otherwise \end{cases} \tag{11}$$

$$\xi_{m,n,l}(y_i, y_j, y_k) = \begin{cases} 1 & y_i = m,\ y_j = n,\ y_k = l \\ 0 & otherwise \end{cases} \tag{12}$$

where $m, n, l \in \{0, 1\}$.

We take the example in Figure 3(b) and Figure 3(c) to illustrate how to generate the factor function $g(y_i, y_j)$ and $g'(y_i, y_j, y_k)$. As shown in Figure 3(b) and 3(c), we can have the following factor functions (Dyadic Friend Relationship):

$$g(y_1, y_3), g(y_2, y_3), g(y_3, y_4), g(y_3, y_5),$$
$$g(y_6, y_7), g(y_6, y_8), g(y_7, y_9), g(y_8, y_9)$$

Also, we can have the following factor functions (Triadic Friend Relationship):

$$g'(y_1, y_2, y_3), g'(y_1, y_2, y_4), g'(y_1, y_2, y_5), g(y_2, y_3, y_4),$$
$$g'(y_2, y_3, y_5), g'(y_3, y_4, y_5), g'(y_6, y_7, y_9), g'(y_6, y_8, y_9),$$
$$g'(y_7, y_8, y_9)$$

Besides, if $y_i = 0, y_j = 1, y_k = 0$, then $\xi(y_i, y_j, y_k) = (0, 0, 1, 0, 0, 0, 0, 0)$. Vector $\psi(y_i, y_j)$ can be calculated in the same way.

C. MODEL LEARNING

Learning the factor graph model is to estimate a parameter configuration $\theta = \{\alpha, \beta, \gamma\}$, so that the log-likelihood of observation information (labeled user pairs) are maximized. By combing Eq.(5), Eq.(7) and Eq.(8), the joint probability defined in Eq.(3) can be rewritten as:

$$P(Y|UPG) = \frac{1}{Z}\exp[\sum\nolimits_i \alpha^{\mathrm{T}}\varphi(\mathrm{x}_i, y_i) + \sum\nolimits_{\|_{i,j}} \beta^{\mathrm{T}}\psi(y_i, y_j) + \sum\nolimits_{\Delta_{i,j,k}} \gamma^T \xi(y_i, y_j, y_k)] \tag{13}$$

One challenge for learning the PIFGM model is that the input data is partially-labeled. To calculate the global normalization term Z, one needs to sum up the likelihood of possible states for all nodes in UPG, 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 inferring from the known labels. We can define the log-likelihood objective function as $L(\theta)$:

$$\begin{aligned} &L(\theta) \\ &= \log p(Y^L|UPG) \\ &= \log \sum\nolimits_{Y|Y^L} p(Y|UPG) \\ &= \log \sum\nolimits_{Y|Y^L} \frac{1}{Z}\exp[\sum\nolimits_i \alpha^{\mathrm{T}}\varphi(\mathbf{x}_i, y_i) + \sum\nolimits_{\|_{i,j}} \beta^{\mathrm{T}}\psi(y_i, y_j) \\ &\quad + \sum\nolimits_{\Delta_{i,j,k}} \gamma^T \xi(y_i, y_j, y_k)] \\ &= \log \sum\nolimits_{Y|Y^L} \exp[\sum\nolimits_i \alpha^{\mathrm{T}}\varphi(\mathbf{x}_i, y_i) + \sum\nolimits_{\|_{i,j}} \beta^{\mathrm{T}}\psi(y_i, y_j) \\ &\quad + \sum\nolimits_{\Delta_{i,j,k}} \gamma^T \xi(y_i, y_j, y_k)] - \log Z \\ &= \log \sum\nolimits_{Y|Y^L} \exp[\sum\nolimits_i \alpha^{\mathrm{T}}\varphi(\mathbf{x}_i, y_i) + \sum\nolimits_{\|_{i,j}} \beta^{\mathrm{T}}\psi(y_i, y_j) \\ &\quad + \sum\nolimits_{\Delta_{i,j,k}} \gamma^T \xi(y_i, y_j, y_k)] - \log \sum\nolimits_{Y} \exp[\sum\nolimits_i \alpha^{\mathrm{T}}\varphi(\mathbf{x}_i, y_i) \\ &\quad + \sum\nolimits_{\|_{i,j}} \beta^{\mathrm{T}}\psi(y_i, y_j) + \sum\nolimits_{\Delta_{i,j,k}} \gamma^T \xi(y_i, y_j, y_k)] \end{aligned} \tag{14}$$

To solve the objective function, we can consider a gradient decent method. Specifically, taking the parameters α as an example, we can calculate the gradients as:

$$\begin{aligned} \frac{\partial(L(\theta))}{\partial\alpha} &= \frac{\sum_{Y|Y^L} \exp\alpha^T \sum\varphi(\mathrm{x}_i, y_i) \cdot \sum\varphi(\mathrm{x}_i, y_i)}{\sum_{Y|Y^L} \exp\alpha^T \sum\varphi(\mathrm{x}_i, y_i)} \\ &\quad - \frac{\sum_{Y} \exp\alpha^T \sum\varphi(\mathrm{x}_i, y_i) \cdot \sum\varphi(\mathrm{x}_i, y_i)}{\sum_{Y} \exp\alpha^T \sum\varphi(\mathrm{x}_i, y_i)} \\ &= \mathrm{E}_{p(Y|Y^L)} \sum\nolimits_{\varphi(\mathrm{x}_i, y_i) - \mathrm{E}_{p(Y)} \sum_{\varphi(\mathrm{x}_i, y_i)}} \end{aligned} \tag{15}$$

Similarly,

$$\frac{\partial(L(\theta))}{\partial\beta} = \mathrm{E}_{p(Y|Y^L)} \sum\nolimits_{\|_{i,j}} \psi_{m,n}(y_i, y_j) - \mathrm{E}_{p(Y)} \sum\nolimits_{\|_{i,j}} \psi_{m,n}(y_i, y_j) \tag{16}$$

$$\frac{\partial(L(\theta))}{\partial\gamma} = \mathrm{E}_{p(Y|Y^L)} \sum\nolimits_{\Delta_{i,j,k}} \xi(y_i, y_j, y_k) - \mathrm{E}_{p(Y)} \sum\nolimits_{\Delta_{i,j,k}} \xi(y_i, y_j, y_k) \tag{17}$$

We take the components in Eq.(16) for example, other components can be explained in the same way. The component $\mathrm{E}_{p(Y|Y^L)} \sum_{\|_{i,j}} \psi_{m,n}(y_i, y_j)$ denotes the expectation of the summation of a dyadic friend relationship feature, given the label distribution over all the relationships conditioned on the labeled relationships. The component $\mathrm{E}_{p(Y)} \sum_{\|_{i,j}} \psi_{m,n}(y_i, y_j)$ denotes the expectation of the same feature given the label distribution over all the relationships. We rewrite the component $\mathrm{E}_{p(Y)} \sum_{\|_{i,j}} \psi_{m,n}(y_i, y_j)$ as:

$$\begin{aligned} \mathrm{E}_{p(Y)} \sum\nolimits_{\|_{i,j}} \psi_{m,n}(y_i, y_j) &= \sum_{Y} p(Y) \sum\nolimits_{\|_{i,j}} \psi_{m,n}(y_i, y_j) \\ &= \sum\nolimits_{\|_{i,j}} \sum\nolimits_{y_i, y_j} \psi_{m,n}(y_i, y_j) p(y_i, y_j) \end{aligned} \tag{18}$$

Another challenge here is that the graphical structure in PIFGM can be arbitrary and may contain cycles, which makes it intractable to directly calculate the marginal probabilities in Eq. (18). We utilize Loopy Belief Propagation (LBP) [31] for approximate calculation due to its ease of implementation and effectiveness. LBP is one popular approximate algorithm to calculate the marginal probabilities in a graphical structure. At the beginning, we approximate marginal probabilities $p(y_i)$, $p(y_i, y_j)$ and $p(y_i, y_j, y_k)$ using LBP, and perform LBP again to obtain those marginal probabilities conditioned on

Algorithm 1 Learning PIFGM

Input: user-pair-oriented graph (UPG=(U_{UPG}, E_{UPG})), Y^L, learning rate η
Output: learned parameters $\theta = \{\alpha, \beta, \gamma\}$
1 Initialize $\theta \leftarrow 0$;
2 Repeat
3 Calculate each $p(y_i, y_j)$ using LBP;
4 Calculate $E_{p(Y)} \sum_{\|i,j} \psi_{m,n}(y_i, y_j)$;
5 Calculate each $p((y_i, y_j)|Y^L)$ conditioned on the observed labels using LBP;
6 Calculate $E_{p(Y|Y^L)} \sum_{\|i,j} \psi_{m,n}(y_i, y_j)$;
7 Calculate the gradient of θ according to Eq.(15-17);
8 Update parameter θ with the learning rate η: $\theta_{new} = \theta_{old} - \eta \cdot \nabla_\theta$
9 until Convergence;

the observed labels. With the marginal probabilities, the gradient can be obtained according to Eq.(15-17). Finally with the gradient, we update each parameter with a learning rate η. The learning algorithm is summarized in Algorithm 1. The components in step(3) $\sim$ step(6) are calculated for the parameter β. The components for the parameters α and γ can also be calculated in the similar way.

Based on the learned parameters, we can obtain the unlabeled user pairs by finding a label configuration that maximizes the joint probability using LBP, i.e.,

$$Y^* = \arg\max_{Y|Y^L} p(Y|UPG) \tag{19}$$

It has to be noted that we use LBP again to calculate the marginal probability of $p(y_i|Y^L, UPG)$ and then predict the label with largest marginal probability.

D. DISTRIBUTED LEARNING

For the expectation calculation in Algorithm 1 (see step(3)), we should calculate the marginal probability $p(y_i)$, $p(y_i, y_j)$, $p(y_i, y_j, y_k)$ by LBP. For step(4), we also should calculate the marginal probability $p(y_i)$, $p(y_i, y_j)$, $p(y_i, y_j, y_k)$ conditioned on the observed labels by LBP. LBP is an iterative process and needs to enumerate all possible label configurations for all the relationships. In Algorithm 1, the most time-consuming step is the gradient calculation. Due to the LBP process, it is important for the learning algorithm to scale up well with large networks. In our paper, we develop a distributed learning method based on MPI (Message Passing Interface) and adopt a master-slave architecture. To divide original PIFGM into different slave processors, we can partition PIFGM into K roughly equal parts, where K is the number of slaves. However, the edges (factors) between different subgraphs are eliminated, which would cause a precision decrease. To solve this problem, we save the whole PIFGM in an adjacent matrix, and then distribute the rows of the adjacent matrix into different machine nodes. In this way, no information is lost, since the dyadic and the triadic friend relationships are all kept in the same machine node. The detailed steps for our method are listed as follows:

Step 1: Divide the UPG into different batches. Save the whole graphical model of UPG in an adjacent matrix and then distribute the rows of the adjacent matrix into different slave nodes.

Step 2: Initialize the parameter θ;

Step 3: The master node sends the newest parameters θ to all the slaves;

Step 4: Slave nodes perform LBP on the corresponding batch to calculate the marginal probabilities, then compute the parameter's gradient and send it back to the master.

Step 5: The master node collects and sums up all gradients obtained from different batches, and updates parameters by the gradient descent method.

Step 6: Goto Step 3, until convergence.

E. CANDIDATE USER PAIR SELECTION

As shown above, the UPG is generated based on the set of candidate user pairs. However, the generated UPG might be very large in practice. Here, we introduce a method based on attributes and structure information for the selection of candidate user pairs. Firstly, we calculate the username similarity to generate seed matching. We define a similarity function for usernames, as shown in Eq.(6). We then add the user pairs into the candidate set X when the similarity is above a threshold. We empirically set the threshold as 0.8. Then, we propagate the matching along with neighborhood relationships, and iteratively add new pairs in the the candidate set X. These new pairs have at least r neighboring pairs already mapped[2]. For example, as shown in Figure 1, we assume that the user pairs (u_1^A, u_1^B) and (u_4^A, u_4^B) have been discovered by username similarity calculation, and we set $r = 2$. Then, we should add four pairs (u_2^A, u_2^B), (u_2^A, u_3^B), (u_3^A, u_2^B), (u_3^A, u_3^B) into the the candidate set X. In our experiment, we associate each pair $(u_i^A, u_j^B) \in (U^A \times U^B)$ with a counter, denoted as M_{ij}. At each iteration time t, we select one pair in $X(t-1)$ and add one to the counter of their neighboring pairs. The user pair having more than r marks is augmented into $X(t)$. The process iterates until there are no more unselected pairs. The pseudocode for the method can be seen in Algorithm 2.

V. EXPERIMENTAL RESULTS

We verified PIFGM on two dataset collections (Section 5.1). All the experiments were conducted using a Linux server with with 128 G memory and 2.0 GHz CPU (Intel Xeon E5-2620 @ 2.00 GHz, 24 cores). The distributed learning algorithm was implemented on MPI (Message Passing Interface).

To quantitatively evaluate the proposed model, we consider the following performance metrics: if a method can find a matching between the two networks, we say that the method correctly recognizes a matching; otherwise, we say that the method makes a wrong recognition. We evaluate

[2]We empirically set *r=4* for SNS dataset, and *r=3* for SR dataset.

Algorithm 2 Candidate User Pair Selection

Input: SMN^A, SMN^B
Output: Candidate user pair set X
1 **For** each $(u_i^A, u_j^B) \in (U^A \times U^B)$ **do**
2 Calculate username similarity $sim(n_i, n_j)$;
3 $X(0) = \{(u_i^A, u_j^B) | sim(n_i, n_j) > T\}$; // $X(t)$ denotes the set of candidate user pair in time t;
4 $t \leftarrow 1$;
5 $R(0) = \emptyset$; // $R(t) \in X(t)$ denotes the set of candidate user pairs that have been selected until time t;
6 $M = 0$; // M_{ij} denotes a counter for the pair (u_i^A, u_j^B);
7 Repeat
8 Randomly select a pair $(u_i^A, u_j^B) \in X(t)$;
9 $M_{i'j'} \leftarrow M_{i'j'} + 1$ where $u_{i'}^A \in N(u_i^A)$, $u_{j'}^B \in N(u_j^B)$;
10 $X(t) = X(t-1) \cup \{(u_{i'}^A, u_{j'}^B)\}$ where $M_{i'j'} > r$;
11 $R(t) = R(t-1) \cup (u_i^A, u_j^B)$;
12 $t \leftarrow t + 1$;
13 **until** $X(t) \backslash R(t) = \emptyset$;

the comparison methods in terms of Precision, Recall, and F1-Measure. They are defined as:

Recall = correct identified users/total identical users

Precision = correct identified users/total identified users

F1 measure = 2*Recall*Precision/(Recall + Precision)

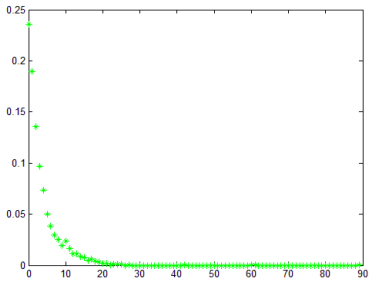
Higher recall rate, precision and F1-measure indicate better performance of a user identification solution.

A. DATASET

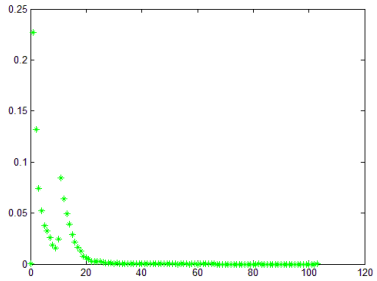
We perform experiments on two collections: SNS dataset [10] and SR dataset (Sina microblog and Renren Network). The SNS data collection consists of five popular online social networking sites: Twitter (TW), Live-Journal (LJ), Flickr (FL), Last.fm (LA), and MySpace (MS). The detailed information about the SNS dataset can be found in [10]. Sina Weibo is one of the most popular Chinese micro-blogging websites, and Renren is a leading real-name social networking Internet platform in China. The Sina Microblog dataset can be captured from the Sina Microblog search page, and consists of $1.21*10^5$ users. The RenRen dataset can be directly obtained from its Open API, and consists of $5.4*10^5$ users. To evaluate the performance of our method, we extract three pairs of graphs from the Sina Microblog dataset and the Renren dataset. The profile features used for both two networks contain gender, birthday, location and educational background. Each pair of graphs is extracted by starting with the identical users and extracting two-layer friends. In each pair of graphs, we manually annotate some identical user pairs. The detailed information about the datasets is listed in Table 3. As shown in Table 3, the SR1 dataset consists of 4238 Sina Microblog users and 11874 Renren users. Because it is difficult to know the exact number of all identical user pairs, we randomly select 200 identified users from the results to calculate the precision. Fig. 5 illustrates the degree distributions of the two networks in SR1. Clearly, both Sina Microblog and Renren follow a power-law distribution.

TABLE 3. SR dataset.

# pair of graphs		(Nodes)	Average degree	Labeled identical user pairs
SR1	Sina	4238	3.19	152
	RenRen	11874	5.42	
SR2	Sina	4329	3.32	143
	RenRen	14735	5.39	
SR3	Sina	5121	3.23	167
	RenRen	16511	5.53	



(a) Sina Microblog(SR1)



(b) Renren(SR1)

FIGURE 5. Degree distributions of SR1 dataset.

B. COMPARISON METHODS

We compare the following methods for user identification across different networks.

(1) SVM: This method formalizes the user identification problem as a binary classification problem. It trains a classification model based on the labeled data, and then adopts the classification model to classify whether a candidate user pair is identical or not. It uses local features defined in Section **4.1** for training the classification model. we use an SVM (www.csie.ntu.edu.tw/∼cjlin/libsvm) as the classifier.

(2) COSNET [10]: A novel energy-based model to formalize our problem as a unified optimization framework. This method considers both local (local attributes, network

structure) and global consistency among multiple networks. We remove global consistency in our experiments, since we evaluate the problem across two networks. We also use the local attributes described in Section 4.1.

(3) JLA [27]: This method is proposed for user profile matching based on Conditional Random Fields that extensively combines usage of profile attributes and social linkage. We use different attributes for different datasets to calculate profile-distance(v, $\mu(v)$), where $v \in SMN^A$ and its projection $\mu(v) \in SMN^B$. For example, profile-distance(v, $\mu(v)$) is calculated as the Levenshtein distance on name matching for SNS network collection. We train a classifier based on C4.5 with MultiBoosting to find incorrect projections.

(4) FRUI [7]: The FRUI solves the user identification problem based on friend relationship. It reveals prior user matched pairs through a limited number of profiles. After a set of prior user matched pairs are identified, a set of new user matched pairs are recognized using network structure in the iteration process. We randomly select 5% labeled pairs as prior user pairs for SNS, and select 100 labeled pairs as prior user pairs for SR.

(5) FRUI-P [8]: The FRUI-P first extracts the friend feature of each user in an SN into friend feature vector, and then calculates the similarities of all the candidate identical users between two SNs. Finally, a one-to-one map scheme is developed to identify the users based on the similarities. Several parameters in this method, such as the positive sample set S, the similarity threshold of the identified users λ, are set as default values.

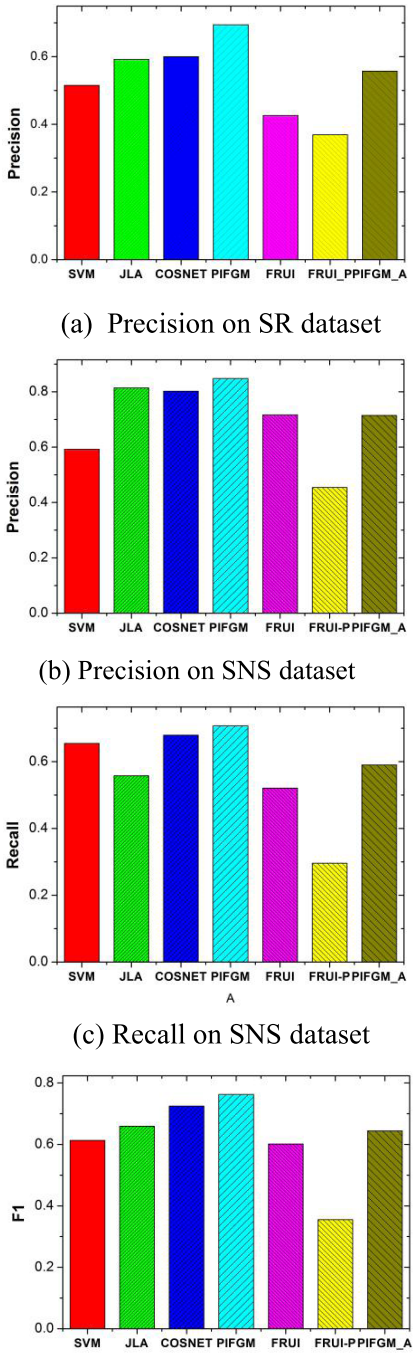
(6) PIFGM_A: To compare with FRUI and FRUI-P, we evaluate a variant of PIFGM by removing the profile attribute factor function $f(x_i, y_i)$, which is denoted as PIFGM_A.

C. USER IDENTIFICATION PERFORMANCE

We perform experiments on both data collections: SNS and SR. For SNS, we partition the ground truth mappings into four groups and conduct four-fold cross validation and report the average results. For SR, we manually label some identical user pairs, and only calculate the precision since the exact number of identical user pairs is unknown.

Figure 6 shows the overall performance of the comparison methods on the two datasets. Clearly, our proposed method achieves better performance than the other comparison methods. In terms of F1-score, PIFGM achieves a 10∼15% improvement over SVM and JLA. PIFGM performs much better than COSNET. This demonstrates that our method is useful for user identification task and is superior to other jointly modeling methods. PIFGM_A also demonstrates its superiority over FRUI and FRUI-P.

We also report the evaluation results for each pair of networks. For each metric, the best results are highlighted in bold. Table 4 lists the detailed performance of hybrid supervised learning methods on each pair of networks. As shown in Table 4, PIFGM achieves the best performance in most tasks. COSNET performs better than JLA and SVM. JLA



(a) Precision on SR dataset

(b) Precision on SNS dataset

(c) Recall on SNS dataset

(d) F1 on SNS dataset

FIGURE 6. Performance of different methods on two collections.

suffers from low recall. This is because that this method propagates information from projections with similar attributes and facilitates discovering other matches. Thus, it can not discover the user pairs with low attribute similarity.

Table 5 lists the detailed performance of structure based methods. To compare with FRUI and FRUI-P, network structure based methods, PIFGM degrades to PIFGM_A. Both PIFGM_A and FRUI adopt friend relationship to propose a uniform network structure based user identification solution. The performance of PIFGM_A is superior to that of FRUI and FRUI-P on all network pairs. In FRUI, the more closely

TABLE 4. Performance comparison of different user identification methods on SNS dataset.

Network pair	SVM			JLA			COSNET			PIFGM		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
LiveJournal-Flickr	0.583	0.772	0.664	0.831	0.502	0.626	0.813	0.773	0.792	0.839	0.781	**0.808**
LiveJournal-Last.fm	0.451	0.603	0.516	0.802	0.632	0.707	0.756	0.703	0.728	0.782	0.712	**0.745**
LiveJournal-Myspace	0.432	0.581	0.496	0.814	0.517	0.632	0.822	0.649	0.725	0.861	0.647	**0.738**
Flickr-Last.fm	0.773	0.612	0.683	0.862	0.526	0.653	0.822	0.652	0.727	0.888	0.674	**0.766**
Flickr-Myspace	0.671	0.532	0.593	0.843	0.472	0.605	0.871	0.593	0.705	0.901	0.636	**0.745**
Last.fm-Myspace	0.701	0.553	0.618	0.781	0.563	0.654	0.811	0.593	0.685	0.844	0.643	**0.729**
Twitter-LiveJournal	0.609	0.723	0.661	0.793	0.624	0.698	0.760	0.751	0.755	0.807	0.768	**0.787**
Twitter-Flickr	0.632	0.721	0.673	0.794	0.523	0.631	0.802	0.724	0.761	0.826	0.734	**0.777**
Twitter-Last.fm	0.503	0.714	0.590	0.813	0.629	0.709	0.783	0.731	0.756	0.803	0.724	**0.761**
Twitter-MySpace	0.571	0.732	0.637	0.805	0.587	0.679	0.782	0.693	0.734	0.827	0.712	**0.765**

TABLE 5. Performance comparison of structure-based methods on SNS dataset.

Network pair	FRUI			FRUI_P			PIFGM_A		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
LiveJournal-Flickr	0.693	0.593	0.639	0.474	0.312	0.376	0.721	0.638	**0.677**
LiveJournal-Last.fm	0.673	0.562	0.613	0.465	0.251	0.326	0.705	0.608	**0.653**
LiveJournal-Myspace	0.739	0.458	0.565	0.485	0.341	0.400	0.716	0.544	**0.618**
Flickr-Last.fm	0.742	0.521	0.612	0.421	0.356	0.386	0.728	0.546	**0.624**
Flickr-Myspace	0.779	0.478	0.592	0.404	0.292	0.339	0.781	0.524	**0.627**
Last.fm-Myspace	0.691	0.498	0.579	0.432	0.329	0.373	0.695	0.532	**0.603**
Twitter-LiveJournal	0.705	0.569	0.630	0.451	0.303	0.362	0.658	0.627	**0.642**
Twitter-Flickr	0.727	0.540	0.620	0.424	0.321	0.365	0.705	0.624	**0.662**
Twitter-Last.fm	0.719	0.531	0.611	0.492	0.214	0.298	0.741	0.615	**0.672**
Twitter-MySpace	0.698	0.459	0.554	0.491	0.239	0.321	0.694	0.641	**0.666**

matched the known friends of a candidate user pair, the higher the chance that they belong to the same real identity, which would result in a low recall rate. The performance of FRUI-P appears not satisfactory. This may be because that a part of the users in these real world SMNs only possess one friend whose features are virtually impossible to be learned from their friends. We deepen the analysis and find that FRUI-P can increase the precision performance on those users with a high number of friends. It suggests that the friend features of users with few friends can hardly been learned. Besides, the precision will increase if we increase the the similarity threshold of the identified users λ, which would also result in the decrease of *recall*.

The above performance demonstrates that our method considers both dyadic friend relationship and triadic friend relationship can identify more potential identical users. PIFGM performs better than PIFGM_A, which demonstrates that the hybrid modeling of attribute matching and network structure matching can help improve matching performance.

Table 6 illustrates the precision results on SR dataset. The precision of PIFGM is around 70%, outperforming other methods in all the three experiments. FRUI-P performs poor over SR. Although this method does not require prior knowledge, the poor performance shows that it is not proper to identify user pairs depending only on it. This findings reveal that PIFGM is much more proficient for recognizing identical users across Sina Microblog and Renren. Besides, PIFGM_A also shows its superiority over FRUI and FRUI-P on both two datasets.

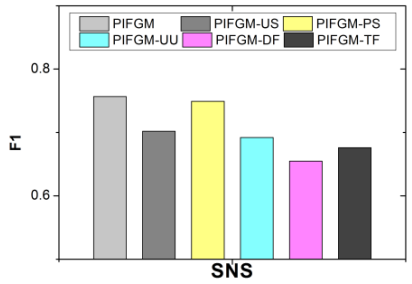
D. ANALYSIS AND DISCUSSION

1) FACTOR CONTRIBUTION ANALYSIS

In PIFGM, we consider multiple factor functions: attribute factor function and friend relationship factor function.

TABLE 6. Performance comparison of different user identification methods on SR dataset.

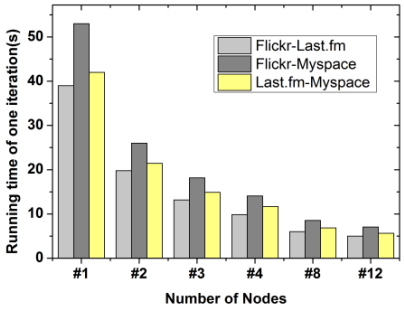
Network pair	Precision						
	SVM	JLA	COSNET	PIFGM	FRUI-P	FRUI	PIFGM_A
SR1	0.529	0.592	0.603	**0.674**	0.392	0.431	0.572
SR2	0.502	0.583	0.572	**0.697**	0.332	0.392	0.503
SR3	0.514	0.602	0.625	**0.713**	0.382	0.456	0.597



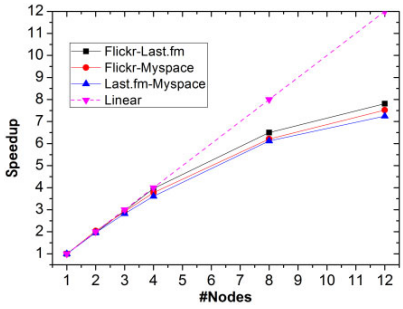
(a) F1 score on SNS dataset

(b) Precision on SR dataset

FIGURE 7. Contribution of different factor functions.



(a)Running time vs. #Cores



(b) Speedup vs. #Cores

FIGURE 8. Efficiency performance.

The former contains three local features: username similarity (US), profile content similarity (PS) and username uniqueness (UU). The latter contains two friend relationships: dyadic friend relationship (DF) and triadic friend relationship (TF). Here we perform an analysis to evaluate the contribution of the different factor functions in our model. We remove one of the five features from the model, and evaluate the deterioration in its performance. Figure 7 shows the F1 score and the precision on the SNS dataset and the SR dataset, respectively. PIFGM-US denotes that we remove the username similarity factor. We can observe clear drop on the performance when removing one of the factors. This indicates that our method works well by combining the different factor functions and each factor in our method contributes to the performance. However, we see that different local factors contribute differently in the different data collections. Username similarity contributes less on SR than on SNS. This is probably because that most usernames in Renren are real names, while in Sina Microblog are not. Dyadic friend relationship features and triadic friend relationship features are helpful to improve F1 score in SNS dataset, but not that helpful to improve precision on SR dataset. We deepen the analysis and find that structure-based factors can increase the recall rate since they are useful to find out more potential identical user pairs that have low attribute similarity.

2) EFFICIENCY PERFORMANCE

We now evaluate the efficiency performance of our distributed learning algorithm on the SNS data set. Since the network pairs in SNS have apparently difference in computing time, we select three network pairs (Flickr-Last.fm, Flickr-MySpace, Last.fm-MySpace), with similar amount of users, to show the running time and the speedup of the distributed algorithm with different number of computer nodes (cores). As shown in Figure 8, it is noticeable that when the number

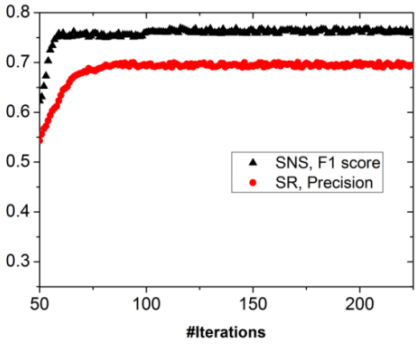


FIGURE 9. Convergence analysis.

of cores increases, the running time of one iteration would decrease apparently. The speedup curve is very close to the linear shape when using 1∼4 cores. Although the speedup inevitably decreases when the number of cores increases, it can achieve 7∼8 speedup with 12 cores. This results demonstrate that the proposed distributed method typically achieves a significant reduction of CPU time.

3) CONVERGENCE ANALYSIS

We analyze the convergence property of PIFGM. We use the average F1-score on ten network pairs in SNS dataset, and the average precision on three network pairs in SR dataset to measure the overall performance. The results in Figure 9 demonstrate that PIFGM can converge quickly within about 100 iterations on both two datasets.

VI. CONCLUSION

This paper studies how to identify users across different social networks. We precisely define the problem, and propose a novel factor graph model-based method PIFGM to address it. We formalize friend relationship into two types: dyadic friend relationship and triadic friend relationship. Then, we incorporate the user profiles' attributes, dyadic friend relationship and triadic friend relationship into a factor graph model. We design several experiments for hybrid method comparison and friend relationship based method comparison. Experimental results on two real-world datasets show that the proposed method significantly outperforms comparison methods. To further scale up to large networks, we propose a distributed learning algorithm based on MPI without information lose. Experiments demonstrate good parallel efficiency of the distributed learning algorithm. However, during the distributed learning algorithm, the speedup inevitably decreases when the number of cores increases. Thus, we will further investigate the fast learning of PIFGM model to reduce the computation time. Furthermore, it would be interesting to investigate how the identified user information can help other applications such as community detection, trust relationship inferring, and link recommendation.

Identifying anonymous users across multiple SMNs is still a challenging work. To achieve a more accurate result, we can incorporate content-based methods into this method, since profile-based, structure-based and content-based methods are complementary and not mutually exclusive. However, the content features (including temporal information, spatial information and posts) are extremely difficult to obtain for the SNS dataset and Renren dataset in our current work. Although not all identical users can be recognized with this method, it can build the foundation for further studies on this issue. Therefore, we suggest applying these methods synergistically for proper decision making purpose in our future work.

REFERENCES

[1] J. Liu, F. Zhang, X. Song, Y.-I. Song, C.-Y. Lin, and H.-W. Hon, "What's in a name?: An unsupervised approach to link users across communities," in *Proc. 6th ACM Int. Conf. Web Search Data Mining (WDM)*, Feb. 2013, pp. 495–504.

[2] R. Zafarani and H. Liu, "Connecting users across social media sites: A behavioral-modeling approach," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2013, pp. 41–49.

[3] X. Mu, F. Zhu, E.-P. Lim, J. Xiao, J. Wang, and Z.-H. Zhou, "User identity linkage by latent user space modelling," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1775–1784.

[4] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Nov. 2013, pp. 179–188.

[5] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: Theory and validation," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 707–719.

[6] T. Man, H. Shen, S. Liu, X. Jin, and X. Cheng, "Predict anchor links across social networks via an embedding approach," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, Jul. 2016, pp. 1823–1829.

[7] X. Zhou, X. Liang, H. Zhang, and Y. Ma, "Cross-platform identification of anonymous identical users in multiple social media networks," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 411–424, Feb. 2016.

[8] X. Zhou, X. Liang, X. Du, and J. Zhao, "Structure based user identification across social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1178–1191, Jun. 2018.

[9] P. Jain, P. Kumaraguru, and A. Joshi, "'@i seek 'fb.me': Identifying users across multiple online social networks," *Proc. 22nd Int. Conf. World Wide Web Companion*, May 2013, pp. 1259–1268.

[10] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "COSNET: Connecting heterogeneous social networks with local and global consistency," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1485–1494.

[11] N. Bennacer, C. N. Jipmo, A. Penta, and G. Quercini, "Matching user profiles across social networks," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, Thessaloniki, Greece, 2014, pp. 424–438.

[12] S. Bartunov, A. Korshunov, S. Park, W. Ryu, and H. Lee, "Joint link-attribute user identity resolution in online social networks," in *Proc. 6th SNA-KDD Workshop*, Aug. 2012, pp. 1–9.

[13] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manils, "How unique and traceable are usernames," in *Privacy Enhancing Technologies*. Cham, Switzerland: Springer, 2011, pp. 1–17.

[14] R. Zafarani and H. Liu, "Connecting corresponding identities across communities," in *Proc. 3rd Int. AAAI Conf. Web Logs Social Media*, Jul. 2009, pp. 354–357.

[15] M. Motoyama and G. Varghese, "I Seek You: Searching and Matching Individuals in Social Networks," in *Proc. 11th Int. Workshop Web Inf. Data Manage.*, Nov. 2009, pp. 67–75.

[16] O. Goga, D. Perito, H. Lei, R. Teixeira, and R. Sommer, "Large-scale correlation of accounts across social networks," Int. Comput. Sci. Inst., Berkeley, CA, USA, Tech. Rep. TR-13-002, 2013.

[17] F. Abel, E. Herder, G.-J. Houben, N. Henze, and D. Krause, "Cross-system user modeling and personalization on the social Web," *User Model. User-Adapted Interact.*, vol. 23, pp. 169–209, Apr. 2013.

[18] Y. Li, Z. Zhang, Y. Peng, H. Yin, and Q. Yu, "Matching user accounts based on user generated content across social networks," *Future Gener. Comput. Syst.*, vol. 83, pp. 104–115, Jun. 2018.

[19] J. Vosecky, D. Hong, and V. Y. Shen, "User identification across multiple social networks," in *Proc. 1st Int. Conf. Networked Digit. Technol.*, Jul. 2009, pp. 360–365.
[20] J. Novak, P. Raghavan, and A. Tomkins, "Anti-aliasing on the Web," in *Proc. 13th Int. Conf. World Wide Web*, May 2004, pp. 30–39
[21] E. Raad, R. Chbeir, and A. Dipanda, "User profile matching in social networks," in *Proc. 13th Int. Conf. Netw.-Based Inf. Syst.*, Sep. 2010, pp. 297–304.
[22] S. Vosoughi, H. Zhou, and D. Roy, "Digital stylometry: Linking profiles across social networks," in *Proc. Int. Conf. Social Inform.*, 2015, pp. 164–177.
[23] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proc. 30th IEEE Symp. Secur. Privacy (SSP)*, May 2009, pp. 173–187.
[24] M. Wang, Q. Tan, X. Wang, and J. Shi, "De-anonymizing social networks user via profile similarity," in *Proc. 3rd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2018, pp. 889–895.
[25] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks," *Proc. VLDB Endowment*, vol. 7, no. 5, pp. 377–388, Jan. 2014.
[26] A. Malhotra, L. Totti, W. Meira, Jr., P. Kumaraguru, and V. Almeida, "Studying user footprints in different online social networks," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 1065–1070.
[27] S. Liu, S. Wang, and F. Zhu, "Structured learning from heterogeneous behavior for social identity linkage," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 7, pp. 2005–2019, Jul. 2015.
[28] H. Fu, A. Zhang, and X. Xie, "Effective social graph deanonymization based on graph structure and descriptive information," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 4, p. 49, Aug. 2015.
[29] E. Kazemi, S. H. Hassani, and M. Grossglauser, "Growing a graph matching from a handful of seeds," *Proc. VLDB Endowment*, vol. 8, no. 10, pp. 1010–1021, Jun. 2015.
[30] C. Chiasserini, M. Garetto, and E. Leonardi, "De-anonymizing clustered social networks by percolation graph matching," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 2, p. 21, Mar. 2018.
[31] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proc. 15th Conf. Uncertainty Artif. Intell.*, Aug. 1999, pp. 467–475.
[32] F. Heider, *The Psychology of Interpersonal Relation*. Hoboken, NJ, USA: Wiley, 1958.
[33] H. Huang, J. Tang, L. Liu, J. Luo, X. Fu, X, "Triadic closure pattern analysis and prediction in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3374–3389, Dec. 2015.

LIDONG WANG was born in Wenzhou, Zhejiang, China, in 1982. She received the Ph.D. degree from the College of Computer Science and Technology, Zhejiang University, in 2013.

She is currently an Associate Professor with Hangzhou Normal University. Her current research interests include image processing, machine learning, and text mining.

KEYONG HU received the Ph.D. degree in mechatronic engineering from the Zhejiang University of Technology, Hangzhou, China, in 2016. He is currently a Teacher of electronic information engineering with the Qianjiang College of Hangzhou Normal University. His research interests include artificial intelligence and new energy technology.

YUN ZHANG received the B.S. and M.S. degrees in computer science from Hangzhou Dianzi University, Hangzhou, in 2006 and 2009, respectively, and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, in 2013. He is currently an Associate Professor with the Zhejiang University of Media and Communications, Hangzhou. His research interests include image and video editing and computer graphics.

SHIHUA CAO received the M.S. degree in computer science from the School of Soft Engineering, Beijing University of Posts and Telecommunications. He is currently a Professor with Hangzhou Normal University. His current research interests include RFID research and artificial intelligence.