Social Identity Link Across Incomplete Social Information Sources Using Anchor Link Expansion

Yuxiang Zhang[1(✉)], Lulu Wang[2], Xiaoli Li[3], and Chunjing Xiao[1]

[1] Civil Aviation University of China, Tianjin, China
{yxzhang, cjxiao}@cauc.edu.cn
[2] Beijing Jiaotong University, Beijing, China
llwang_l4@bjtu.edu.cn
[3] Institute for Infocomm Research, A*STAR, Singapore, Singapore
xlli@i2r.a-star.edu.sg

Abstract. Social link identification SIL, that is to identify accounts across different online social networks that belong to the same user, is an important task in social network applications. Most existing methods to solve this problem directly applied machine-learning classifiers on features extracted from user's rich information. In practice, however, only some limited user information can be obtained because of privacy concerns. In addition, we observe the existing methods cannot handle huge amount of potential account pairs from different OSNs. In this paper, we propose an effective SIL method to address the above two challenges by expanding known anchor links (seed account pairs belonging to the same person). In particular, we leverage potentially useful information possessed by the existing anchor link, and then develop a local expansion model to identify new social links, which are taken as a generated anchor link to be used for iteratively identifying additional new social link. We evaluate our method on two most popular Chinese social networks. Experimental results show our proposed method achieves much better performance in terms of both the number of correct account pairs and efficiency.

Keywords: Social networks · Social Identity Link · Hometown inference

1 Introduction

Online social networks (OSNs), such as Twitter, Facebook, Sina Weibo, Renren and Foursquare, have become more and more popular in recent years. Each social network can be represented as an individual graph and focuses on a specific application. Oftentimes, people are getting involved in numeric social networks concurrently. For example, we can access the latest news from Twitter and Sina Weibo, post our photos using Facebook and Renren, and share interesting places (or locations) with our friends using Foursquare. Thus, it comes as no surprise that many users often have multiple separate accounts in different OSNs, although there are no direct correspondences or connections among these multiple accounts belonging to the same users from different networks.

J. Bailey et al. (Eds.): PAKDD 2016, Part I, LNAI 9651, pp. 395–408, 2016.
DOI: 10.1007/978-3-319-31753-3_32

Discovering the correspondences between accounts of the same user, i.e. social identity link (SIL) problem, by integrating information from multiple OSNs is a crucial prerequisite for many practical Web based applications, such as detecting more accurate community structures [1], finding rising stars in social networks [2], and providing better customer support and personalized services matching the user preferences. For example, if we know a user's Twitter account, then its social connections and location data in Twitter can be used to better recommend the taste to this user in the Foursquare. However, existing research (such as [3–8]) have showed that it is very challenging to identify user accounts of the same natural person across different social media platforms. The main reason is that users and social platform operators take extremely strict measures to avoid divulging user personal information.

Previous studies (such as [7, 9, 10]) assume that they can collect rich user information/attributes about *user profiles, user generated content, behaviors and friend networks*. After collecting all the rich attributes for each user from different social networks, existing methods mainly employ supervised learning techniques [3, 7, 8, 11–13] (with an exception which uses unsupervised learning [14]) to build binary classification models for SIL prediction. However, it is very difficult, if not impossible, to obtain user's private information in many real-world applications. As such, existing research will suffer when only incomplete information is available.

The second facing challenge is that current classification methods are not feasible to handle huge amount of potential account pairs from different OSNs. Particularly, the computational cost for identifying pair-wise accounts is N_1*N_2 (N_1 and N_2 are the number of accounts in source and target networks, respectively). We can imagine how many account pairs could be generated given each OSN could have more than 1 billion users (e.g. Facebook). Clearly, it will be extremely time consuming, if not impossible, to perform the intensive classification task.

In this paper, we employ open APIs, provided by the social platform operators, to only collect the *publically available* attributes, including 6 user profile attributes, such as nickname, gender, birthday, university name, university entry year and location, and friend network attribute. Thus, we are handling the SIL problem in a difficult but *practical* scenario with *incomplete* information sources. In addition, we also observe that many profile attributes have missing or false values, making this research even more challenging. Additionally, to tackle the second challenge, contrast to existing standard classification methods, we leverage anchor link information and propose a local search strategy to iteratively identify the new social links. Our proposed approach largely reduces the search space and is thus more feasible than existing methods for handling those real-world large scale OSNs.

2 Related Works

SIL problem across different social platforms has been studied in recent few years. User link was formalized as connecting identity problem across communities in [3–6] in the early stage. Subsequently, various methods were proposed.

The performance of existing methods largely relies on the extracted features, from user profiles, user generated content, behaviors and friend networks. Some research

papers [15–19] heavily focus on username parsing to link multiple online identities of a user, based on the assumption that same users will have the similar names from different social platforms. Paper [20] studies three features extracted from the content created by a user, i.e. timestamp of posts, geo-location attached to post and writing styles. It finds that the geo-location of posts is the most powerful features to identify social links. Another research explores the social meta path concept (which is a means to capture connection information in the social networks) to generate useful compound features from friendship, location, timestamp of post and post content [21]. Some works have shown that other information about users, like their group memberships [22] and tagging behavior [23], can also be used to uniquely identify users. More recent papers [7, 10, 14, 24] have integrated as many features as possible to identify social links across different social networks, since researchers believe that less features are not sufficient enough to achieve good performance.

Unfortunately, in practice we can only obtain the limited information, leading to limited or incomplete features and thus much worse results. In addition, the existing methods are also inefficient and the computational costs are prohibitively high, as they need to classify large amount of all the possible account pairs from different networks. In this paper, we leverage those potentially useful information possessed by the anchor link to overcome the above two weaknesses from the existing methods.

3 Overall Algorithm

Denote P as the set of all natural *persons* in real life. For a social network G, represent $V(G)$ as the set of all *accounts,* each belonging to a distinct user. An injective function ϕ_G: $V(G) \rightarrow P$ maps each account in $V(G)$ to a natural person in P.

Social Identity Link, SIL. Given an account I_i^S from a source network G^S (i.e. $I_i^S \in V(G^S)$), social identity link problem is to find a corresponding account I_j^T from a target network G^T(i.e. $I_j^T \in V(G^T)$), such that $\phi_S(I_i^S) = \phi_T(I_j^T)$. This definition is very strict. In fact, formula should be associated with a certain probability or confidence score.

Firstly, we need to collect a seed anchor link set *ALS*, consisting of the account pairs where one account from source anchor set AR^S in G^S and the other account is from target anchor set AR^T *in* G^T: $ALS = \{(ar_i^S, ar_i^T) | (ar_i^S, ar_i^T)$ is an anchor link provided, $ar_i^S \in AR^S$, $AR^S \subseteq V(G^S)$, $ar_i^T \in AR^T$, $AR^T \subseteq V(G^T)\}$. *ALS* can be obtained by either questionnaires or rule-based filtering methods.

Secondly, starting from an anchor link from *ALS*, our proposed the anchor link local expansion algorithm *iteratively* searches the new putative social identity links until they cannot be found. Figure 1 shows the key idea of our proposed method. Given an anchor link (ar_i^S, ar_i^T) (ar_i^S and ar_i^T are the anchor nodes from source or target network respectively), we first visit 3^S that is any one of neighbors of the account ar_i^S, and then we try to find a best matching account from G^T. If $1'^T$ is found, the new social link, called *generated* anchor link, $(3^S, 1'^T)$, can be leveraged to further identify other social links. Thus, a set of social links is identified in the order of the following sequence $(ar_i^S, ar_i^T) \rightarrow (3^S, 1'^T) \rightarrow (5^S, 4'^T) \rightarrow (4^S, 5'^T)$.

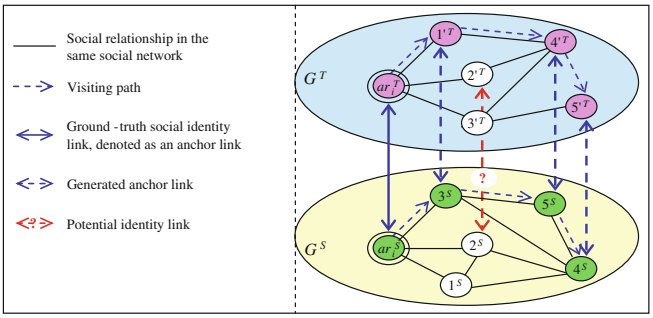

Fig. 1. Key idea overview.

This detailed algorithm is shown in Algorithm 1. In step 1, we initialized a queue Q as an empty set, which will be used to store the given anchor nodes in AR^S or newly generated anchor nodes from source network. We also initialize an output set O. Steps 2 and 3 mark all the nodes in the source network as "unvisited". From steps 4 to 12, we will go through all the anchor nodes ar_i in AR^S and generated anchor nodes. Particularly, for each unvisited anchor node in the source network (step 8), we first find all its neighbors (step 9). Then, for each of the neighbors, function *FindSIL*() in step 10 is used to find the best match account in the target network – the detailed process will be introduced in next section. Step 11 will add the newly generated anchor nodes from the source network into the queue Q. Finally, the results are returned in the step 13.

Algorithm 1. Overall Algorithm

Input: *source network* G^S; *target network* G^T; *anchor link set ALS*
Output: *matched account pairs in O*

1: *Initialize a queue* $Q=\emptyset$; $O=\emptyset$ *//Initialize a queue Q and set O;*
2: **For** (*int* $j=0$; $j<|V(G^S)|$; $j++$) *//For all the accounts in* G^S
3: $Mark[v_j]=unvisited$;
4: **For** (*int* $i=0;i<|AR^S|;i++$) *//For all the anchors in* AR^S
5: ***Q.enqueue***(ar_i); // $ar_i \in AR^S$
6: **While** (*Q.empty*())
7: $u=Q.pop()$;
8: **If** ($Mark[u]==unvisited$)
9: **While** ($k<|N(u)|$) // $N_k(u)$ *is* k^{th} *neighbor of u.*
10: **If** ($l=$***FindSIL***$(N_k(u))$) *//FindSIL() is used to find the best match account of* $N_k(u)$ *in target network* G^T.
11: $O = O \cup \{(N_k(u), l)\}$
12: ***Q.enqueue***$(N_k(u))$
13: Return O as the set of matched account pairs

The worst-case time complexity of Algorithm 1 is $O(|V(G^S)||E(G^S)|)$ time, where $|V(G^S)|$ and $|E(G^S)|$ are the number of accounts and the number of edges between users in source network G^S respectively.

The intra-network connections associated with the anchor link are very useful. In particular, given an anchor link (ar_i^S, ar_i^T) and ar_i^S's neighbor I_i^S, we believe the best matching account I_j^T is likely to locate in a small range of the anchor account ar_i^T. This is because, the friend connections in one network have higher chance to be re-occurred in another network either directly (re-occurrence connections) or indirectly (friends' friends). As such, our proposed technique can largely reduce the search space based on this novel idea.

4 Optimal Search Range

Our goal is to solve SIL problem with minimum expected computational cost. We define search range and use it to control computational cost in the target network G^T.

We first define the shortest distance between two nodes u and v, i.e. $d(u, v)$, as the number of edges in the shortest paths. Let $P(u, v)$ be the set of all paths that start from u and end at v. Note $d(u, v)$ is ∞ if v is not reachable from u:

$$d(u,v) = \begin{cases} \arg\ \min_{p \in P(u,v)} |p| & \text{if} \quad P(u,v) \neq \varnothing \\ \infty\ \text{otherwise.} \end{cases} \tag{1}$$

Search Range. Given an anchor link (ar_i^S, ar_i^T) where ar_i^S and ar_i^T are from network G^S and G^T respectively, the search range $R_d(ar_i^T)$ around ar_i^T in G^T is defined by $R_{d \leq n}(ar_i^T) = \{I^T \in G^T \mid d(ar_i^T, I^T) \leq n\}$.

Here, n is a non-zero natural number. In the best case, d is equal to 1; that is to say, $R_{d=1}(ar_i^T)$ represents a set of *direct* neighbors of ar_i^T. At worst case scenario, d is less than or equal to infinity; that is to say, $R_{d \leq inf}(ar_i^T)$ represents a set of all accounts in G^T. *Our strategy* for selecting the search range is to gradually grow the value of the d from 1 to infinity according to specific requests. This strategy can largely reduce the search space. The effect of the parameter d on the system performance of user identification is discussed in the Subsect. 6.2 in detail.

5 Identity Matching

We introduce how to select a best match account from the candidate set. Particularly, we first define some distinguishing features in nickname, hometown and friend network. Learning models are subsequently used to find the best match accounts.

5.1 Features Definition

(1) **Nickname Similarity:** Features derived from the nickname have been widely used to identify the social links across different social platforms. There are even a few studies, such as [8, 11], which only use nickname features for identification.

However, in many real datasets, there are too few consistent names (namesakes) across different social platforms. In our dataset, 98 % of ground-truth linked account

pairs (which is created manually for the same users from different social platforms.) possess different nicknames. Nonetheless, we find that many pairs of different nicknames belonging to same users are somewhat related. We have performed comparison of these ground-truth social links and summarized the most frequent relationships between two nickname pairs as follows: (1) there exists a common substring, such as (张金鹏, zjp金鹏042); (2) a common substring occurs many times in one nickname, such as (辛倩文, 小辛辛辛辛辛); (3) there are no differences if Chinese characters are converted into alphabets, such as (范一真, 范熠禎) (both are Fan Yizheng). In order to tackle the case (3), we convert Chinese characters into their corresponding alphabets when there are mismatches between two nicknames written in Chinese characters.

Before calculating nickname features, we introduce some basic notations and definitions. We denote the nickname of an account by $Ni(.)$ for two accounts I_i^S and I_j^T, and $p = |Ni(I_i^S) \cap Ni(I_j^T)|$ is thus the size of *common/overlapping characters* between $Ni(I_i^S)$ and $Ni(I_j^T)$. A function $lcs(.)$ is to compute *the longest continuous common substring* between two nicknames, which is implemented by the generalized suffix tree [25]. We define $q = |lcs(Ni(I_i^S), Ni(I_j^T))|$ as the length of the longest continuous common substring between $Ni(I_i^S)$ and $Ni(I_j^T)$. We use r and s to represent the *frequency* which $lcs(Ni(I_i^S), Ni(I_j^T))$ occurs in $Ni(I_i^S)$ and $Ni(I_j^T)$, and in *all* nicknames respectively. Finally, function $len(.)$ and $max(.)$ are used to compute the *length* of a nickname, and the *maximum* nickname length.

Finally, the nickname similarity $NiS(I_i^S, I_j^T)$ between $Ni(I_i^S)$ and $Ni(I_j^T)$ is defined as follows: $NiS(I_i^S, I_j^T) = (CoC(Ni(I_i^S), Ni(I_j^T)) + LoS(Ni(I_i^S), Ni(I_j^T)) + ReS(Ni(I_i^S), Ni(I_j^T)) + SpS(Ni(I_i^S), Ni(I_j^T)))/4$, where $CoC(Ni(I_i^S), Ni(I_j^T)) = p/max(len(Ni(I_i^S)), len(Ni(I_j^T)))$ is used to reflect the contribution from the *common characters*. $LoS(Ni(I_i^S), Ni(I_j^T)) = q/max(len(Ni(I_i^S)), len(Ni(I_j^T)))$ is used to reflect the contribution of the *longest common substring*. $ReS(Ni(I_i^S), Ni(I_j^T)) = r/(len(Ni(I_i^S)) + len(Ni(I_j^T)))$, on the other hand, is used to reflect the contribution of the *repetitions of the longest common substring*. $SpS(Ni(I_i^S), Ni(I_j^T)) = 1/\sqrt{s}$, is used to reflect the contribution of *rarity of the longest common substring* in all nicknames. Typically, those account pairs with less rare longest common substring will get higher similarity than those frequent ones, as they are more helpful for identification purpose.

(2) **Hometown Similarity:** We observe that different social networks could have different types of location information. Sina Weibo and Twitter only possess the current location, while Renren and Facebook possess many different types of locations, such as hometown, current city and workplace. Because hometown in Renren may be different from current location in Sina Weibo for same users (we could move to other places for education or to make a living), we are facing a very challenging task, i.e. how to compute hometown/location similarity based on different types of location information.

Our two interesting observations help us to tackle this challenging problem. One is that the hometown is still the same as the current location for some account pairs. For example, in our ground-truth linked account pairs, there are 46 % of account pairs, of which location in Sina Weibo is the same as hometown in Renren (they are kind of permanent dwellers in their hometown). The other is that,

for some accounts, some of their friends have been re-occurred in multiple networks, like a mirror. According to these two phenomena, we propose the following solution to this task.

Given an account I_i^S in G^S and an account I_j^T in G^T, we denote hometown of I_i^S and I_j^T by $Hi(I_i^S)$ and $Ho(I_j^T)$ respectively. The probability score $HoS(I_i^S, I_j^T)$ that $Hi(I_i^S)$ is equal to $Ho(I_j^T)$ can be computed as follows: $HoS(I_i^S, I_j^T) = P(Hi(I_i^S) = Ho(I_j^T))$. As mentioned above, Sina Weibo as the source network only has the current location. Although Renren has the hometown, some accounts do not fill up the hometown and some accounts may fill up the false hometown, to make our problem even more difficult. In other words, we are not sure whether the value of a hometown is true or not. As such, we cannot match the value of $Hi(I_i^S)$ and value of $Ho(I_j^T)$ directly. In this paper, we propose a novel hometown inference model by leveraging the location information of neighbors.

For an account I_i^S in G^S, we denote the set of neighbors of I_i^S by $N(I_i^S)$, and denote the set of current location of accounts in $N(I_i^S)$ by $CL(N(I_i^S))$. Likewise, we use $N(I_j^T)$ to represent the set of the neighbors of I_j^T in G^T. The set of hometown of accounts in $N(I_j^T)$ is denoted by $HT(N(I_j^T))$. The intersection of sets $CL(N(I_i^S))$ and $HT(N(I_j^T))$ is denoted by $CH(I_i^S, I_j^T) = CL(N(I_j^T)) \cap HT(N(I_i^S))$, and let the size of $CH(I_i^S, I_j^T)$ be K.

In addition, the value of hometown of I_j^T is denoted as l_h, which may be either empty or filled up by user. If l_h has been filled up, we should then take into account the contribution of l_h to the hometown similarity even though we can not make sure whether l_h is true or not. For this reason, we define a new hometown set $HT_1(N(I_j^T)) = \{HT(N(I_j^T)), l_h\}$. Let $CH_1(I_i^S, I_j^T)$ be the intersection of $CL(N(I_i^S))$ and $HT_1(N(I_j^T))$.

We consider six different cases for the location and hometown mapping, illustrated in Fig. 2. These cases represent different relationships among $CL(N(I_i^S))$, $HT(N(I_j^T))$ and l_h. The probability score $HoS(I_i^S, I_j^T)$ will be computed according to each specific case. The weight of each edge is the frequency of the hometown/current.

Case (1) shown in Fig. 2(a). The intersection $CH_1(I_i^S, I_j^T)$ is empty, i.e., $CH_1(I_i^S, I_j^T) = \emptyset$. We can derive $HoS(I_i^S, I_j^T) = 0$.

Case (2) shown in Fig. 2(b). The value of hometown of I_j^T is empty and the $CH(I_i^S, I_j^T)$ is not empty, i.e., $l_h = \emptyset \wedge CH(I_i^S, I_j^T) \neq \emptyset$. The $HoS(I_i^S, I_j^T)$ is computed by the formula: $HoS(I_i^S, I_j^T) = \sum_{k=1}^{K} P(Hi(I_i^S) = l_k) P(Ho(I_j^T) = l_k)$, where $l_k \in CH(I_i^S, I_j^T)$. For example, we can compute $HoS(I_i^S, I_j^T) = 0.1 \times 0.3 + 0.6 \times 0.4 = 0.27$.

Case (3)–(4) shown in Fig. 2(c)–(d). The value of hometown of I_j^T is not empty, the $CH_1(I_i^S, I_j^T)$ is not empty, and l_h does not appear in $CH_1(I_i^S, I_j^T)$, i.e., $l_h \notin \emptyset \wedge CH_1(I_i^S, I_j^T) \neq \emptyset \wedge l_h \notin CH_1(I_i^S, I_j^T)$. The $HoS(I_i^S, I_j^T)$ is computed by the following formula: $HoS(I_i^S, I_j^T) = \sum_{k=1}^{K} P(Hi(I_i^S) = l_k) P(Ho(I_j^T) = l_k) + aP(Ho(I_j^T) = l_h)$, where $l_k \in CH_1(I_i^S, I_j^T)$, and a is the weight of the additional account-attribute relationship from I_i^S to l_h, and assigned to the minimum of all weights related to I_i^S. The weight a is used to reflect the contribution of l_h to the hometown similarity. For example, a is equal to 0.1 in Fig. 2(c). So $HoS(I_i^S, I_j^T)$ is equal to $0.3 \times 0.3 + 0.6 \times 0.6 + (0.1 \times 0.1) = 0.46$.

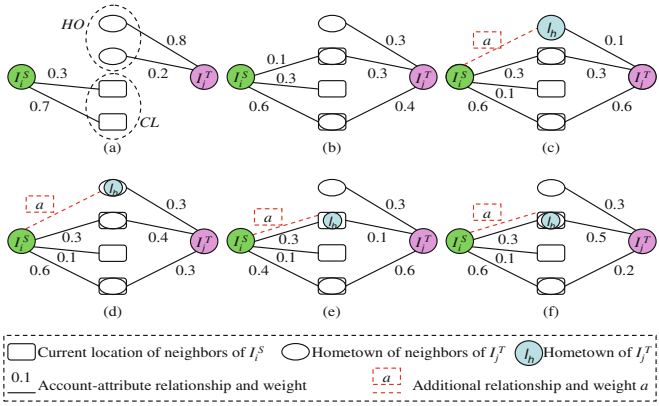

Fig. 2. Different cases tackled in hometown inference.

Case (5)–(6) shown in Fig. 2(e)–(f). The value of hometown of I_j^T is not empty, the $CH_1(I_i^S, I_j^T)$ is not empty, and l_h appears in $CH_1(I_i^S, I_j^T)$, i.e., $l_h \notin \emptyset \wedge CH_1(I_i^S, I_j^T) \neq \emptyset \wedge l_h \in CH_1(I_i^S, I_j^T)$. The $HoS(I_i^S, I_j^T)$ is computed by the formula (5), in which a is used to reflect the contribution of l_h to the hometown similarity, and is equal to $P(hi(I_i^S) = l_h)$. For example, a is equal to 0.1 in Fig. 2(*e*). $HoS(I_i^S, I_j^T)$ is equal
to 0.3.

(3) **Friendliness**: We suppose that accounts ar_i^S and ar_i^T belong to the same user, and I_i^S is a neighbor of ar_i^S. If the degree of friendliness between I_j^T and ar_i^T is high, we believe that I_i^S and I_j^T likely belong to the same user. The triadic closure principle [26] can be used to indirectly explain this underlying inference.

Because the search range $R_d(ar_i^T)$ constrains the friendliness score $FrS(ar_i^T, I_j^T)$, which is related to the parameter d. Then, $FrS(ar_i^T, I_j^T)$ can be evaluated by the following metrics: (1) $FrS_1(ar_i^T, I_j^T) = 1$, $d \geq 1$; (2) $FrS_2(ar_i^T, I_j^T) = |CN(ar_i^T, I_j^T)|$, $d \geq 2$; (3) $FrS_3(ar_i^T, I_j^T) = |CN(N(ar_i^T), I_j^T)|$, $d \geq 3$. Here $CN(.)$ represents the common neighbors between two accounts, $|CN(.)|$ is the size of $CN(.)$.

5.2 Decision Model on Pairwise Similarity

(1) **Machine Learning Models:** Existing studies on the social identity link identification mainly rely on the supervised classification model. There are four classification models, namely multilayer perceptron (MLP) in [4], support vector machine (SVM) in [7, 14], logistic regression (LR) in [8], and Naive Bayes (NB) in [15], which have been demonstrated to perform well for this problem. As such, we also build these four classification models using our labeled dataset so that we can apply them to select the best match account from the candidate set in the target network. The experiments are described in detail in Subsect. 6.2. The results reported in Fig. 3 show that LR and MLP models are more accurate. Thus, we select LR and MLP models for our experiments on the whole dataset.

(2) **Algorithm for Finding the Best Match Account:** After all problems mentioned above have been solved, we integrate all solutions into the algorithm *FindSIL*(), which is used for finding the best match account I_j^T *of* I_i^S. Note the detailed *FindSIL*() algorithm is shown in Algorithm 2, which is called in our overall Algorithm 1.

Algorithm 2. FindSIL()

Input: (1) *account* I_i^S*, which is from* G^S *and waiting for identification;* (2) *anchor links* (ar_i^S, ar_i^T); (3) *parameter d, which is used to control the search range* $R_d(ar_i^T)$.
Output: *best match account* I_j^T
1: *Define the search range* $R_d(ar_i^T)$ *according to* ar_i^T *and d.*
2: *Find the best match account from the candidates through the decision model.*
3: *Return identified account* I_j^T.

6 Experiments

6.1 Experimental Setup

(1) **Data Preparation:** As there is no publicly available benchmark datasets for social identity link, we have to create our own data sets for performance evaluation purpose. We leverage two publicly available large-scale social network data sets from China for our experiments. One is Sina Weibo dataset, and the other one is Renren dataset.

Before crawling user profile datasets from the two social networks for account linking, we make sure that the profiles of the linked users have overlaps, at least partially. In this paper, we request that the crawled accounts must satisfy a constraint, i.e. their university profile from two social platforms is equal to a specific value. We crawled 40,618 Renren accounts and 20,448 Sina Weibo accounts. The number of average friends per account in Renren and Sina Weibo is 339.9 and 27.5, respectively.

(2) **Evaluation Metrics:** We conduct our experiment on both the small set of labeled data and the large set of unlabeled data, i.e. those nodes in the target network to be identified. The objective of the former is to determine the best classification models, while the objective of the latter is to identify as many social links as possible.

For the first experiments on the small set of labeled data, we evaluate the effectiveness of various methods, using *precision*, *recall* and *F-score*, which are standard metrics in machine learning and information retrieval, and have widely been used for user identification across different social networks [7, 19, 22].

For the second experiments on the large set of unlabeled data, we need to manually check each of predicted linked account pairs, which can be classified into three categories: *correct* account pairs (*tp*)*, uncertain* account pairs (*up*) and *wrong* account pairs (*fp*). Let the total number of predicted account pairs be $total = tp + up + fp$. The correct ratio or precision (*Pr*), uncertain ratio (*Ur*) and

wrong ratio (*Wr*) are computed as follows: $Pr = tp/total$, $Ur = up/total$, $Wr = fp/total$. In addition, we evaluate the proportion of predicted account pairs in whole data set. We define the *coverage ratio* (*Cr*) as $Cr = total/min(N_1, N_2)$, where N_1 and N_2 are the number of accounts in the source network and target network, respectively.

(3) **Comparative Methods:** In this subsection, we compare our proposed methods with the following state-of-the-art methods.

(1) *Nickname Similarity Method* (*NSM*): Many features extracted from nicknames have been used to predict the social links. Especially, a few studies [8, 11] *only* use the nickname similarity features. We also implement a *NSM* method (only use the nickname features in Subsect. 5.1 (1)) to predict social link.

(2) *Rule-based Filtering Method* (*RFM*): The rule-based filtering method uses hand-picked similarity features and rules designed to predict identity link. This method achieves good performance [19]. We build a prototype *RFM* system based on this paper, which has won the second prize in the third China Software Developing Contest in 2014 (www.cnsoftbei.com).

(3) *Our Method based on Logistic Regression* (*OM-LR*): we use LR model to select the best match account from the search range.

(4) *Our Method based on Multilayer Perceptron* (*OM-MLP*): we use MLP model to select the best match account from search range.

Note that the performance of our proposed *OM-LR* and *OM-MLP* methods is related to the parameter d in the search range $R_d(ar_i^T)$.

6.2 Experimental Results

We first aim to find the best classification models through performing experiments on the small set of labeled data. Here, the labeled dataset consists of 1,304 positive and some negative instances where the number of negative instances is determined by an imbalance ratio and the number of positive instances.

We partition the dataset into two groups using 10-fold cross validation (CV). Note this is different from standard CV as we use less training data and more test data, to

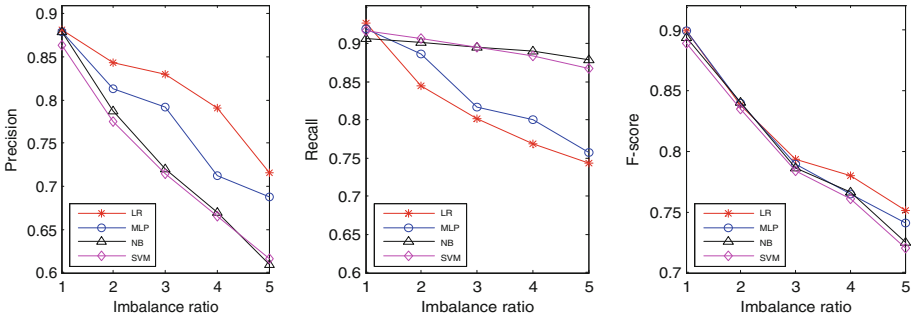

Fig. 3. Performance comparison of different classification models under imbalance ratios.

better reflect the real scenario. We report the average results of 10-fold CV. In each iteration of the cross validations, we sample the negative account pairs according to different imbalance ratios. Figure 3 shows the performance comparison of 4 different classification models under 4 different imbalance ratios (1:5). From Fig. 3, LR and MLP get better performance than the other two methods, and their performance is very close. So we choose both LR and MLP models for selecting the best match account from candidates.

Secondly, we compare our methods, OM-LR and OM-MLP, with existing *NSM* and *RFM* methods, based on the large set of unlabeled ground-truth data. As the objective of this experiment is to identify as many social links as possible, the search range includes all accounts from the target network ($d = inf$). Our methods use 186 randomly selected anchor links only and Table 1 shows experimental results for different methods. The first four columns show the performance in terms of various evaluation metrics. The fifth column *#coap* denotes the number of *correct* account pairs and the last column *Total* is the total number of predicted account pairs.

From Table 1, the performance of the *NSM* method is worst among all the methods, as it predicted only 316 correct account pairs and with lowest coverage ratio 1.8 %. We observe that rule based method *RFM*, albeit accurate (with highest precision), its coverage ratio Cr = 5.4 %, is much lower than 13.7 % and 13.1 % of our proposed OM-LR and OM-MLP respectively. In addition, the number of correct account pairs identified by *RFM* is much less than that by our OM-LR and OM-MLP. In summary, our methods, especially OM-LR, can identify much more correct social links than existing methods, which cannot be identified by both *NSM* and *RFM* methods.

Table 1. Performance comparison of different methods on the unlabeled data ($d = inf$).

	Pr	Ur	Wr	Cr	#coap	Total
NSM	84.7 %	3.2 %	12.1 %	1.8 %	316	373
RFM	92.6 %	1.7 %	5.7 %	5.4 %	1017	1098
OM-LR	59.6 %	3.5 %	36.9 %	13.7 %	1667	2798
OM-MLP	58.8 %	3.1 %	38.1 %	13.1 %	1576	2673

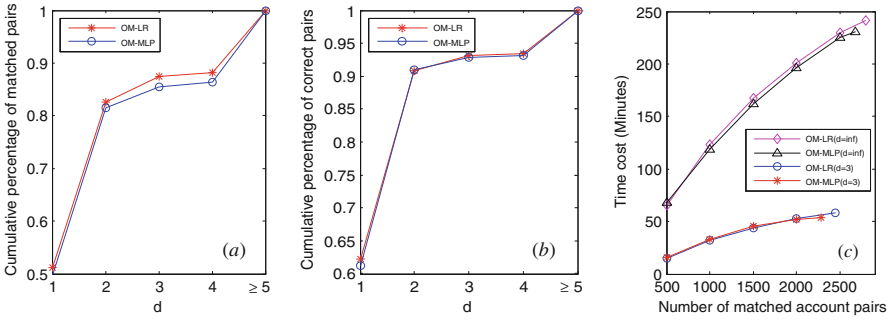

Fig. 4. The cumulative percentage of the matched account pairs (*a*) and of the correctly matched account pairs (*b*) with regard to the different *d*; (*c*) Time cost.

Finally, we evaluate the effectiveness of OM-LR and OM-MLP under different values of parameter d. Fig. 4(*a*) shows the cumulative percentage of the matched account pairs with regard to the different *d*. About 80 % of account pairs are matched with $d \leq 2$ by the OM-LR or OM-MLP. In particular, there are about 50 % of matched account pairs, in which each account from the target network is found in $R_{d=1}(ar_i^T)$; Fig. 4(*b*) shows the cumulative percentage of the correctly account pairs with regard to the different *d*. About 90 % of account pairs are correctly matched with $d \leq 2$ by the OM-LR or OM-MLP. Only about 7 % of account pairs are correctly matched with $d \geq 5$. These statistics reveal that most of correct social links can be found in small range ($d \leq 2$) around their anchor links and generated anchor links. As such, our methods can solve the social link identification efficiently using a local search strategy.

6.3 Efficiency Evaluation

In existing methods, the computational cost of identifying pair-wise accounts is $N_1 * N_2$ (N_1 and N_2 are the number of accounts in G^S and G^T respectively). For our method, the computational cost is estimated as follows.

Given an account I_i^S in G^S and an anchor link (ar_i^S, ar_i^T), the number of accounts in the search range $R_d(ar_i^T)$ of I_i^S is denoted by $|R_d(ar_i^T)|$. Assuming $d \leq m$, $|R_{d\leq m}(ar_i^T)| = |R_{d=1}(ar_i^T)| + \ldots + |R_{d=m}(ar_i^T)|$ ($\cap_{j=1}^{m} R_{d=j}(ar_i^T) = \varnothing$). Denote the corresponding search tree by $Tr(G^T)$, and let the average number of friends per account in $Tr(G^T)$ is k_2', then we can compute the number of accounts of $Tr(G^T)$ by $N_2' = k_2' * ((k_2')^m - 1)/(k_2' - 1) \approx (k_2')^m$. Obviously, N_2' is much smaller than N_2 when the depth parameter m is small.

Let us consider real social networks described in the Subsect. 6.1. The average shortest path length of RenRen is about 5 (that was also confirmed by the work [27]). Assume $m = 4$, which is less than the actual value. Then the average number of friends per account is $k_2' = 14.2$ in the search tree $Tr(G^T)$. The actual average number of friends per account in G^T is about 340 computed by $k_2 = 2E_2/N_2$, where E_2 is the number of friends. Obviously, there are 325.8 (340–14.2) redundant accounts. Knowing that about 93 % of account pairs are correctly matched with $d \leq 3$ in our methods, the number of accounts in the search range can be estimated by $|R_{d\leq 3}(ar_i^T)| = |R_{d=1}(ar_i^T)| + \ldots + |R_{d=2}(ar_i^T)| = k_2' + (k_2')^2 + (k_2')^3 \leq k_2 + k_2 k_2' + (k_2')^3 = 8{,}031$, which is much less (<20 %) than $N_2 = 40{,}618$. If we assume $m = 5$, then the number of accounts in the search range $|R_{d\leq 3}(ar_i^T)| = 3{,}766$, which is smaller than that $m = 4$.

Next, we also use the total execution time to evaluate the efficiency of different methods. We conduct experiments on our methods with different parameter values, i.e. $d = inf$ and $d \leq 3$, where $d = inf$ represents the computational cost $N_1 * N_2$, while $d \leq 3$ denotes the computational cost $N_1 |R_{d\leq 3}()|$. The parameter $d \leq 3$ is reasonable because about 93 % of account pairs are correctly matched with $d \leq 3$ by our methods. The experiments and latency observations are conducted on a PC, with Intel® Core™ i5-4460 processor and 8 GB main memory.

Figure 4(*c*) shows the relationship between the number of the matched accounts and the time cost. The time cost of OM-LR ($d \leq 3$) and OM-MLP($d \leq 3$) is significantly less than the time cost of OM-LR($d = inf$) and OM-MLP ($d = inf$) for identifying the

same number of account pairs. Of course, the high efficiency of OM-LR($d \leq 3$) and OM-MLP($d \leq 3$) is at the expense of slightly lower *coverage ratio*. Nevertheless, as we handle large-scale networks, it is thus acceptable. In addition, the time cost of OM-LR and OM-MLP with the same d value is very close.

7 Conclusion

In this paper, we address the problem of linking user accounts of the same natural person across different social networks. Our proposed method is based on our unique theoretical assumption inspired by the triadic closure principle. In particular, given two user accounts of the same natural person across different social media platforms, their friends/neighbors in different social platforms should still be directly or indirectly connected to itself. Based on the theoretical assumption, we propose a novel method, which *is to link accounts across different social platforms using the local expansion strategy*. Experimental results demonstrate that our proposed method outperforms existing methods significantly. Note our proposed method is generic and thus it can be applied to link up user accounts across other Chinese or English social networks (e.g. Twitter and Facebook), as long as we can collect their large-scale network data.

Acknowledgments. This work was partially supported by grants from the National Natural Science Foundation of China (Grant No. U1533104, U1333109, 61301245, 61305107).

References

1. Li, X.-L., Foo, C.S., Tew, K.L., Ng, S.-K.: Searching for rising stars in bibliography networks. In: Zhou, X., Yokota, H., Deng, K., Liu, Q. (eds.) DASFAA 2009. LNCS, vol. 5463, pp. 288–292. Springer, Heidelberg (2009)
2. Li, X.-L., Tan, A., Yu, P.S., Ng, S.-K.: ECODE: event-based community detection from social networks. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part I. LNCS, vol. 6587, pp. 22–37. Springer, Heidelberg (2011)
3. Carmagnola, F., Cena, F.: User identification for cross-system personalization. Inf. Sci. **179**(1–2), 16–32 (2009)
4. Vosecky, J., Hong, D., Shen, V.Y.: User identification across multiple social networks. In: Proceedings of NDT 2009 (2009)
5. Zafarani, R., Liu, H.: Connecting corresponding identities across communities. In: Proceedings of ICWSM 2009 (2009)
6. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: Proceedings of S&P (2009)
7. Liu, S., Wang, S., Zhu, F., Zhang, J., Krishnan, R.: HYDRA: large-scale social identity linkage via heterogeneous behavior modeling. In: Proceedings of SIGMOD 2014 (2014)
8. Zafarani, R., Liu, H.: Connecting users across social media sites: a behavioral-modeling approach. In: Proceedings of KDD 2013 (2013)
9. Jain, P., Kumaraguru, P.: Finding nemo: Searching and Resolving Identities of Users across Online Social Networks (2012). arXiv preprint arXiv:1212.6147

10. Jain, P., Kumaraguru, P., Joshi, A.: @i seek 'fb.me': identifying users across multiple online social networks. In: Proceedings of WWW 2013 (2013)
11. Malhotra, A., Totti, L., Meira, W., Kumaraguru, P., Almeida, V.: Studying user footprints in different online social networks. In: Proceedings of Advances in Social Networks Analysis and Mining, 2012 (2012)
12. Nunes, A., Calado, P., Martins, B.: Resolving user identities over social networks through supervised learning and rich similarity features. In: Proceedings of SAC 2012 (2012)
13. Zhang, H., Kan, M.-Y., Liu, Y., Ma, S.: Online social network profile linkage. In: Jaafar, A., Mohamad Ali, N., Mohd Noah, S.A., Smeaton, A.F., Bruza, P., Bakar, Z.A., Jamil, N., Sembok, T.M.T. (eds.) AIRS 2014. LNCS, vol. 8870, pp. 197–208. Springer, Heidelberg (2014)
14. Liu, J., Zhang, F., Song, X., Song, Y.I., Lin, C.Y., Hon, H.W.: What's in a Name?: an unsupervised approach to link users across communities. In: Proceedings of WSDM 2013 (2013)
15. Goga, O.: Matching User Accounts across Online Social Networks: Methods and Applications. Ph.D. thesis, University Pierre and Marie CURIE (2014)
16. Iofciu, T., Fankhauser, P., Abel, F., Bischoff, K.: Identifying users across social tagging systems. In: Proceedings of ICWSM 2011 (2011)
17. Kong, X., Zhang, J., Yu, P.S.: Inferring anchor links across multiple heterogeneous social networks. In: Proceedings of CIKM 2013 (2013)
18. Anwar, T., Abulaish, M.: An MCL-based text mining approach for namesake disambiguation on the web. In: Proceedings of ICWI 2012 (2012)
19. Carmagnola, F., Osborne, F., Torre, I.: User data discovery and aggregation: the CS-UDD algorithm. Inf. Sci. **270**(20), 41–72 (2014)
20. Goga, O., Lei, H., Krishnan, S., Friedland, G., Sommer, R., Teixeira, R.: Exploiting innocuous activity for correlating users across sites. In: Proceedings of WWW 2013 (2013)
21. Zhang, J.W., Yu, P.S., Zhou, Z.H.: Meta-path based multi-network collective link prediction. In: Proceedings of KDD 2014 (2014)
22. Li, J., Wang, G.A., Chen, H.: Identity matching using personal and social identity features. Inf. Syst. Front. **13**(1), 101–113 (2011)
23. Iofciu, T., Fankhauser, P., Abel, F., Bischoff, K.: Identifying users across social tagging systems. In: Proceedings of AAAI Conference on Weblogs and Social Media, 2011 (2011)
24. Chen, Y., Zhuang, C., Cao, Q., Hui, P.: Understanding cross-site linking in online social networks. In: Proceedings of SNA-KDD 2014 (2014)
25. Gusfield, D.: Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press, New York (1999)
26. Rapoport, A.: Spread of information through a population with socio-structural bias i: assumption of transitivity. Bull. Math. Biophys. **15**(4), 523–533 (1953)
27. Zhao, X., Sala, A., Zheng, H., Zhao, B.: Efficient shortest paths on massive social graphs. In: Proceedings of CollaborateCom 2011 (2011)