



Social account linking via weighted bipartite graph matching

Jiangtao Ma^{1,2}  | Yaqiong Qiao¹ | Guangwu Hu^{3,4}  | Tong Li⁵ | Yongzhong Huang¹ | Yanjun Wang¹ | Chaoqin Zhang¹

¹State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

²Zhengzhou University of Light Industry, Zhengzhou 450001, China

³School of Computer Science, Shenzhen Institute of Information Technology, Shenzhen 518172, China

⁴Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China

⁵Huawei Technologies Co., Ltd., Shenzhen 518055, China

Correspondence

Guangwu Hu, School of Computer Science, Shenzhen Institute of Information Technology, 518172, Shenzhen, China.
Email: hugw@sziit.edu.cn

Funding information

Foundation of Henan science and Technology Department, Grant/Award Number: No.162102410076, No.162102310578; Natural Science Foundation of Guangdong Province, Grant/Award Number: No.2015A030310492 2015A030310492; Fundamental Research Project of Shenzhen Municipality, Grant/Award Number: JCYJ20160301152145171; National Natural Science Foundation of China, Grant/Award Numbers: 61402255, 61170292, 61373161 and 61672470; National key Research and Development projects intergovernmental cooperation in science and technology of China, Grant/Award Numbers: 12016YFE0100300 and 12016YFE0100600; Foundation of Henan Educational Committee, Grant/Award Numbers: 17A520064 and 16A520062

Summary

Along with the increasing popularity of online social network (OSN), it is common that the same user holds many accounts among different OSNs (eg, Facebook, Twitter, WeChat, QQ). In this scenario, an interesting and challenging problem arises: how to link accounts among OSNs belonged to a natural person, which is also known as a graph matching problem. The solution helps understand user behaviors and offer better services. To solve the account linking problem, various techniques for OSNs have been proposed. However, most existing methods assume specific OSN features impractical in general OSNs and unscalable to large-scale OSNs. To address these shortcomings, in this paper, we remodel the account linking problem into maximum matching on weighted bipartite graphs and utilize the Kuhn-Munkres algorithm to solve it. In our solution, we capture user profile, user online time distribution, and user interest as features to describe user accounts and measure account similarity, which is used as weight of edge in bipartite graphs. Then, the maximum matching on weighted bipartite graphs is solved with the Kuhn-Munkres algorithm. The experiments conducted on the real datasets show that our solution outperforms the baseline methods with 11%, 17%, and 29% on average in precision, recall, and F1 score, respectively.

KEYWORDS

account linking, bipartite graph matching, node matching, social network, user identification

1 | INTRODUCTION

In recent years, social network analysis and mining have attracted more and more attentions.¹⁻³ By the end of 2016, Facebook was considered as the largest online social

network (OSN) with its 1.86 billion monthly active users, and WhatsApp was the second largest OSN with its 1 billion monthly active users. According to the statistics from China Internet Network Information Center,⁴ there are 731 million netizens using mobile phones or smart

terminal devices to access the Internet by the end of 2016 in mainland China. The top 3 most visited OSNs are WeChat moments (85.8%), QZone (67.8%), and Sina Weibo (37.1%). A study⁵ shows that 82% LinkedIn users and 91% Twitter users also use Facebook to enjoy the services of different OSNs. In order to provide better services, a comprehensive understanding of user information is necessary. Therefore, we need to combine user information across OSNs and generate a comprehensive user portrait. In the process of user information fusion, an important question is how to link user accounts that belonged to the same user across different OSNs. In other words, how to map the nodes between graphs, which is known as the user account matching problem across different OSNs. The solution is helpful in recommending systems,^{6,7} entity linking,⁸ and entity resolution.⁹

Some OSN platforms (eg, Google+) allow users to display their other social accounts in profile pages, from which we can see the relationship between accounts. Some OSN platforms allow the use of telephone numbers or e-mail addresses to retrieve accounts, such as WeChat accounts, which can be searched through QQ number or mobile phone number. It is also possible to utilize user names to link accounts across real-name OSN platforms, eg, Facebook and LinkedIn accounts can be associated with user's real name. A single sign-on network can log in a different OSN through the same ID, which can be associated with the account across OSNs. The previously mentioned methods rely on the OSN platform for technical supports and active participations of users. If the OSN platform itself does not offer this function, or the number of participants is not big enough, these methods will not work efficiently.

Account linking is a link prediction problem across OSNs, which is to find the correspondence between user accounts in different OSNs. Similar to account linking, de-anonymizing user accounts across OSNs is to identify user accounts with auxiliary user accounts from other OSNs. Many studies have addressed this issue with user profile attributes, such as screen names, genders, birthdays, locations, education background, positions, and profile photos. However, profile attributes are easily faked for privacy concerns or impersonated by malicious users. These profile-based methods are quite fragile in these scenarios. Some researchers have employed public user activities to link accounts using writing styles, geo-tags, and timestamps of user generated contents (UGCs). Because these information is offered only in specific OSNs, these techniques have difficulties in scaling to general OSNs. Some researchers have leveraged user topology in social graphs to link accounts across OSNs. However, this method needs lots of matrix computation which has high time complexity.

One of the core assumptions about user account linking is that the behaviors of the same user on different social networking platforms are similar. For example, users always connect with real-world friends, focus on similar topics, and even publish the same content across OSNs. These methods use usernames, personal profiles, UGCs, social relations, and other features. User account linking is converted into a supervised classification problem with these features. These methods work well if the dataset is small, but scalability is poor when encountering large-scale OSNs.

MOBIUS¹⁰ employs usernames to link accounts across social networks. Zafarani et al analyze user's behavioral patterns and extract features to capture information redundancies according to analysis of these patterns. Finally, they build a learning framework to train a learning model. They categorize behavioral patterns into human limitations, exogenous factors, and endogenous factors. The proposed behavioral modeling approach exploits information redundancy according to these behavioral patterns. Different from MOBIUS, we utilize user account profile, user online time distribution, and user interest features to map user accounts across OSNs.

In order to solve the social account linking problem with desirable performance, in this paper, we propose a novel solution with the maximum matching on weighted bipartite graphs through the Kuhn-Munkres (KM) algorithm. To achieve that, we transformed the account linking problem into a maximum matching problem on weighted bipartite graphs. We use user account similarity to link accounts across OSNs. The similarity between accounts is used as the weight of edge in social graphs. We utilize features including user profiles, user's online time distribution pattern, and user interests to link accounts and calculate the similarity between accounts. The KM algorithm is used to solve the maximum matching problem on weighted bipartite graphs to achieve our goals. In summary, our contributions can be summarized as follows:

1. We use features including user profiles, user's active online time distribution, and user interests to measure the similarity of social accounts.
2. We propose a joint learning model to combine these 3 types of features so as to measure the similarity of user accounts and adjust the weights of these features in similar metric social accounts by balancing factors.
3. We propose an account linking method based on the KM algorithm with the maximum matching on weighted bipartite graphs and extend the application of the KM algorithm.
4. We conduct experiments on 4 real network datasets and compare our solution with the state-of-the-art

methods. The experimental results show that our proposals outperform the baseline methods with 11%, 17%, and 29% on average in precision, recall, and F1 score, respectively.

The rest of this paper is organized as follows. Section 2 summarizes the related work. The account linking problem is stated and formulated in Section 3. Our scheme is elaborated in Section 4 and is evaluated and compared with baseline methods in Section 5. At last, Section 6 concludes this paper.

2 | RELATED WORK

The last decade witnesses lots of researches on account linking in OSNs. Various social account linking methods have been proposed. Because de-anonymizing user account across OSNs is closely related to account linking, we mainly reviewed the existing methods from these aspects.

2.1 | Social account linking

An immediate intuition is to use usernames to match accounts across OSNs. Zafarani et al¹¹ are the first to study this problem, and they map user accounts of different online communities by adding and deleting username suffixes. Peritio et al¹² use the Hidden Markov Model to estimate the uniqueness of usernames. Irani et al¹³ find that users often use the deformation of their names as screen names. Motoyama et al¹⁴ first use profile information (such as gender, location, occupation, and university) to link accounts and help new users to find their friends' accounts. The bag-of-words model is used to measure common words among profile attributes. However, this model cannot distinguish words that have similar meanings with minor spelling changes. Malhotra et al¹⁵ use vector to represent profile attributes, and the corresponding dimension represents the corresponding profile attributes. The similarity of the 2 accounts is measured by the cosine similarity of the vector. Although these schemes can achieve a good performance in specific OSNs, they suffer from profile authenticity and data integrity. The reason is users may not provide their authentic and complete profiles for privacy concerns.

Considering user behaviors are unique and cannot be impersonated easily compared with user profiles, researchers start to resort to this feature to link user accounts across OSNs. Zafarani et al¹⁶ associate accounts with user behavior patterns as a primary feature. The following factors are considered when registering a user name: user's knowledge and memory limitation, personal

typing habits, hobby, keyboard layout, and language input habits. A comprehensive analysis of these factors can help users understand the patterns when registering their usernames. Liu et al¹⁷ solve the social account linking problem based on user behavior patterns including user profile attributes, UGCs, and social connections. Mishari et al¹⁸ find that user's writing style can be used as a useful feature for linking user accounts. They use user's language model and writing style to establish a feature model and link user accounts. Kong et al¹⁹ propose a multi-network anchor link method based on user location, time, and text information to identify related accounts in different OSNs. Zhang et al²⁰ link user accounts using the location information involved in UGCs. Liu et al²¹ train a semi-supervised learning model based on user attributes, UGC, social structures, and behavior trajectories. They propose a multi-objective optimization framework to correlate accounts. Nevertheless, the user behavior data availability is harder than before out of privacy concerns, which makes user behavior-based linking methods unable to scale to general OSNs.

Social graph structure is another characteristic of user account. The integration of this feature and other features can improve the performance of account linking. Tan et al²² use hyper-graphs to model social graphs and propose a manifold alignment framework to map user accounts into a common low-dimensional space for account linking. Cui et al²³ study account linking between OSNs and e-mail networks, where profile similarity and social connection similarity are integrated. They also find the account linking patterns between e-mail networks and Facebook. Kong et al¹⁹ convert the account linking problem of Foursquare and Twitter into a stable matching problem of 2 sets of elements and use the link prediction method to infer the linking relation between accounts. Bartunov et al²⁴ present a joint link-attribute method based on conditional random field utilizing profile and network structure features. Koutra et al²⁵ transform the account linking problem into an optimization problem, which is transformed into the optimal permutation function of the 2 adjacency matrices. They use the relative degree distance method to initialize the adjacency matrices and use the stochastic gradient descent method to find the optimal solution. Zhou et al²⁶ design a unified framework for account linking based on network structures. They use the matched accounts as seed accounts and use the seed accounts to iterate and link more user accounts. Zhang et al²⁷ utilize cardinality constraints to solve link prediction problem. They minimize the link loss of transformation between the feature vector and labels. Thus, the link prediction problem is converted into an optimization problem with multiple variables, where cardinality constraints are modeled as mathematical constraints on

node degree. Zhang et al²⁸ employ ego network features and user behavior features to link accounts among multiple social networks. Despite these methods have effective performance in solving accounts linking, but the user social structure-based matching method is difficult to scale to large OSNs because matrix computation time is expensive for sparse OSNs.

2.2 | De-anonymizing social networks

In order to analyze the de-anonymization and privacy problem, Narayanan and Shmatikov²⁹ design an account linking method for Twitter and Flickr, where account linking only based on network topology. Even if the overlapped information between the target network and the secondary network is little, the robustness of this method is good. Similar techniques are utilized to de-anonymize the Netflix data with the IMDB data.³⁰ Narayanan et al³¹ employ the de-anonymization method to do link prediction for the Kaggle dataset, and they utilize the random forest method to do link prediction between nodes. Similarly, Sharad and Danezis³² address the de-anonymization problem using the random forest method to match the node pairs automatically. Korula and Lattanzi³³ use the Erdős-Rényi random graph and preferential attachment model to link accounts from intense nodes (users with a large number of friends). This method proposes a many-to-many mapping algorithm based on the number of unmatched users and the number of common neighbors. In addition, it uses 2 control parameters to fine tune the algorithm performance. Actually, the Erdős-Rényi random graph model is only mathematically meaningful but impractical in OSNs. However, the quantification is effective under an assumption of identified seeds, which is impractical for real-world de-anonymization attacks.

However, these methods cannot utilize some background information such as uncertain user relationships. Backstrom et al³⁴ present active and passive attack methods based on sub-graph search patterns in an anonymous network to learn social relationships between users. However, these methods are effective in identifying relationships between nodes in small-scale OSNs, but not for large-scale OSNs. Zhou and Pei^{35,36} analyze the de-anonymize attack based on neighborhood and propose the k-anonymity and l-diversity methods to protect privacy. Unfortunately, these anonymization techniques are vulnerable when attackers have lots of background knowledges. Wondracek et al³⁷ point out that group membership is sufficient to de-anonymization users in OSNs. Furthermore, Qian et al³⁸ suggest that knowledge graph is an effective model to de-anonymize and infer privacy in OSNs. Ji et al³⁹ de-anonymize users with seed information and present the theoretical foundation for structure-

based de-anonymization attacks. However, these models do not consider that the attacker's auxiliary information might be probabilistic.

In this paper, we focus on solving the social account linking problem with the maximum matching on weighted bipartite graphs. We will state the problem and provide solutions in detail in the following section.

3 | PROBLEM STATEMENT AND FORMULATION

3.1 | Problem statement

To better understand the account linking problem across OSNs, we take Figure 1 as an example. Suppose a user owns an account in each of OSNs, eg, account pair (A, A') belongs to 1 natural person. In short, our aim is to accurately and effectively identify account pairs across OSNs.

However, account linking across OSNs has the following challenges:

1. It is difficult to measure the relationships between accounts across OSNs.
2. It is challenging to measure the similarity of the accounts across OSNs with user profile, user online time distribution, and user interest features.
3. It is challenging to match similar accounts across large sparse OSNs with an effective and efficient method.

3.2 | Problem formulation

The relationship between the accounts across OSNs is the mapping relationship between the vertices in social graphs, and we use Definition 1 to describe the relationships.

Definition 1. Given source OSN $G^s = (V1, E1)$, target OSN $G^t = (V2, E2)$, $V1$ and $V2$ represent the set of user accounts. $E1$ and $E2$ indicate the social relationships between user accounts.

Our goal is to find the correspondence between the account set $V1$ and $V2$, which can be transformed into a bipartite graph matching problem, as defined by Definition 2.

Definition 2. Given a weighted bipartite graph $G = (V = (v_i^s, v_j^t), E)$, $v_i^s \in V1$, $v_j^t \in V2$, E is candidate account linking between v_i^s and v_j^t , the weight of the E is $E_{weight} = Sim(v_i^s, v_j^t)$, and M is the maximum matching of G , if

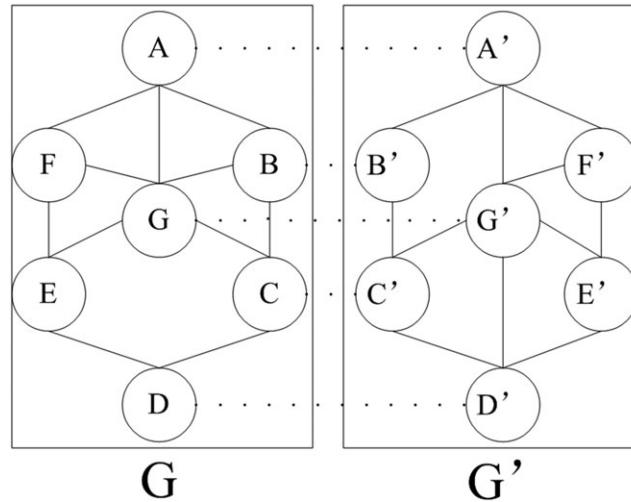


FIGURE 1 Example illustration of the account linking problem. There are 2 OSNs G and G' , and each node, ie, A, A' , in different networks denotes a user account. The dashed lines between nodes in different networks (eg, $A-A', B-B', C-C', D-D'$) are account linking pairs, which indicate the 2 nodes at the ends of the line belong to the same person. Our goal is to find out the account linking pairs among different OSNs with desirable accuracy and precision

$|M| = \min(|V1|, |V2|)$, M is a complete matching, while if $|V1| = |V2|$, M is a perfect matching.

In order to measure the similarity of the v_i^s and v_j^t $Sim(v_i^s, v_j^t)$, we take the similarities of user profile similarity $Sim_{Profile}(v_b^s, v_b^t)$, user online time distribution similarity $Sim_{Time}(v_b^s, v_b^t)$, and user interest similarity $Sim_{Interest}(v_b^s, v_b^t)$ into account.

We can find the account linking across G^s and G^t with Formula 1, which is to find the maximum matching of the weighted bipartite graph G . The constraint is the sum of the similarity value should be the maximum.

$$M = \text{Maxweightbigraphmatch}(G^s, G^t) \quad (1)$$

$$\text{subject to } \text{argmax} \left(\text{sum} \left(\text{sim} \left(v_i^s, v_j^t \right) \right) \right)$$

One way to solve the maximum matching of weighted bipartite graphs is to use the KM algorithm, but it is only suitable for perfect matching of bipartite graphs. Actually, the number of user accounts in 2 OSNs is unequal, so we fabricate a user account v_j^t to make user accounts equal across 2 OSNs. Because G^s and G^t are partially aligned, the alignment between 2 networks is a soft probabilistic alignment, and the nodes are not a 1-to-1 mapping. We add $\|G^s| - |G^t\|$ nodes to the smaller network, and we do not need to specify a priori the number of accounts that we want to match. If the similarity of the 2 accounts is higher than a similarity threshold, the 2 accounts are considered to be a node pair. Otherwise, if an account could not find an account that their similarity

is under similarity threshold, the account is matched to a fabricate account. This matched account pair which contains fabricate account is deleted in the final stage. After deleting the account pair containing fabricate accounts from the perfect matching, we can get the account linking of G^s and G^t .

4 | SCHEME DETAILS

In this section, we describe our scheme in detail. We first present user profile similarity in Section 4.1. The user online time distribution similarity and user interest similarity are presented in Sections 4.2 and 4.3, respectively. After that, we propose a unified framework for incorporating these 3 similarity measure features together in Section 4.4. Section 4.5 gives the account linking details based on the KM algorithm.

4.1 | User profile similarity

It is the simplest and straightforward way to characterize user similarity by using user profile attributes. If 2 user accounts have the same age, gender, occupation, interest, or other information, they may belong to the same user. The bag-of-words model is used to calculate the similarity of 2 user profiles. But this is just a statistical model without semantic information, so the recall rate is low. We first use user profile attributes (such as private phones, e-mail addresses, locations, occupations, universities, genders) to match user accounts. If 2 accounts have the same private telephone number or e-mail address, the 2 accounts should belong to the same person; otherwise,

use other attributes to calculate the profile similarity. The similarity of user account can be calculated by Equation 2:

$$Sim_{profile}(v_i^s, v_j^t) = \sum_{k=1}^m I_k p_k(v_i^s, v_j^t) \quad (2)$$

where m is the number of the attributes, I_k is the weight of attribute importance, $I_k \in [0, 1]$, $\sum_{k=1}^m I_k = 1$. $p_k(v_i^s, v_j^t)$ is the similarity of k^{th} attribute of v_i^s and v_j^t . We use Algorithm 1 to calculate profile similarity. Table 1 shows the details of algorithm 1.

We notice that, in algorithm 1, if the user account has same telephone number or same e-mail address, the similarity of user profile is 1. We employ this kind of accounts as seeds to link accounts across social networks. We extend the topology of these seed accounts with n depth of its neighbors, and we construct 2 networks according to these seed accounts, where n is the depth of depth-first search (DFS). Our goal is find the linking accounts across the 2 networks.

4.2 | User online time distribution similarity

Many OSNs allow users to automatically share UGCs to multiple OSNs simultaneously. For example, the Instagram users could share their UGCs to Instagram, Facebook, Twitter, Flickr, Tumblr, Foursquare, and Sina Weibo. The UGCs have almost identical timestamps, which can be utilized to link corresponding accounts. The distribution of the UGC timestamps is unique for each user. For example, some users prefer tweeting before sleep, while some users prefer sharing UGCs during travel. The time distribution of different accounts can help us find out the relationships between network accounts. Zhang et al¹⁹ propose to use the number of common timestamps of UGCs to measure the active time distribution similarity of 2 accounts. But in reality, the

timestamps of UGCs are not exactly the same due to time delay across OSNs. So, this method cannot accurately measure the active time distribution similarity of the 2 accounts.

However, user's online time distribution is unique, which can be used to measure the active time distribution similarity of 2 accounts. The number of posts during each period can be considered as user activity index; $\mathbf{X} = (x_1, x_2, \dots, x_b, x_{|X|})$ and $\mathbf{Y} = (y_1, y_2, \dots, y_b, \dots, y_{|Y|})$ are the amount of posts by time sequence of v_i^s and v_j^t , respectively. $|\mathbf{X}|$ and $|\mathbf{Y}|$ are the lengths of the sequences of \mathbf{X} and \mathbf{Y} , respectively, $|\mathbf{X}| = |\mathbf{Y}| = n$. We use the Euclidean distance to measure the distance of the time sequences with equal lengths, and the activity distance of X and Y is measured by Equation 3:

$$L(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^2 \right)^{1/2}. \quad (3)$$

The time complexity is $O(n)$. The similarity of the user time activity between v_i^s and v_j^t is calculated by Equation 4:

$$Sim_{Time}(v_i^s, v_j^t) = \frac{1}{1 + L}. \quad (4)$$

However, in practice, when the lengths of 2 time sequences are unequal. For example, a Facebook account published a status, but the corresponding Twitter account did not tweet, the releasing time is not 1-by-1 mapping across OSNs. If $|\mathbf{X}|$ and $|\mathbf{Y}|$ are unequal, the Euclidean distance method cannot work well. The dynamic time warping method can solve the time sequence similarity with different lengths, but the time complexity is $O(N^2)$. We use the fast dynamic time warping method⁴⁰ to measure the distance of the 2 time sequences for its time complexity $O(N)$. We use \mathbf{X} and \mathbf{Y} to build a warping path $\mathbf{W} = w_1, w_2, \dots, w_k$, where k is the

TABLE 1 Algorithmic description of profile similarity calculation

Algorithm 1 profile similarity calculation

Input: Two social networks, $G^s(V1, E1)$ and $G^t(V2, E2)$, $v_i^s \in V1$, $v_j^t \in V2$.

Output: Profile similarity of node v_i^s and v_j^t

1: **For** ($i = 0$; $i < n$; $i++$) // n is the number of shared attributes of user accounts

2: **For** ($j = 0$; $j < n$; $j++$)

3: **If** (v_i^s (tel) == v_j^t (tel) || v_i^s (email) == v_j^t (email))

4: $sim(v_i^s, v_j^t) = 1$;

5: **Else**

6: $sim_{profile}(v_i^s, v_j^t) = \sum_{k=1}^m I_k p_k(v_i^s, v_j^t)$;

7: **Return** $sim(v_i^s, v_j^t)$

length of the warping path, $\max(|X|, |Y|) \leq k < |X| + |Y|$, and $w_k = (i, j)$ is the k^{th} warping path. Therefore, the distance of warping path is represented by Equation 5:

$$Dist(W) = \sum_{k=1}^{k=K} Dist(w_{ki}, w_{kj}). \quad (5)$$

Dynamic programming is used to solve the warping path through Equation 6:

$$D(i, j) = Dist(i, j) + \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} \quad (6)$$

where $D(i, j)$ is the minimum warping distance of 2 time sequences with lengths i and j respectively, and $Dist(i, j)$ is the distance of x_i and y_j . We use $FastDTW(X, Y) = Dist(W)$ as the distance between sequences X and Y , so the user online time distribution similarity of accounts v_i^s and v_j^t is calculated by Equation 7:

$$Sim_{Time}(v_i^s, v_j^t) = \frac{1}{1 + FastDTW(X, Y)}. \quad (7)$$

Therefore, $Sim_{Time}(v_i^s, v_j^t) \in [0, 1]$, if the distance of sequence X and Y is 0, the similarity of the user online time distribution of v_i^s and v_j^t is 1; if the distance of sequence X and Y is $+\infty$, the similarity of the user online time distribution of v_i^s and v_j^t is 0. In the experiments, w is the number of the daily tweet user posts. We set k as 30, which means the monthly activity. W is the sequence of w_1, w_2, \dots, w_k . In most scenarios, user has similar activities across social networks, but the peaks are not aligned across social networks, because a user updates her activities on 1 network and does not update her activities on other networks sometimes. Therefore, we need to warp the activities to align the closed activities before computing the similarity of user activities. DTW can be utilized to calculate the similarity of time sequences by extending or shrinking the activity sequence. The sequence represents the active degree of users on the social networks, which is an important feature of user account.

4.3 | User interest similarity

User interests are helpful to many applications such as online precise marketing, personalized search, and recommendation systems. User interests are stable in a short period of time because birds of a feather flock together. Users prefer making friends with those who share the same opinions or interests. At the same time, users are more willing to accept the ideas they believe. Some OSN users will show their interests and arguments in real life through their behaviors in OSNs. Therefore, user interests can be employed to link user account across OSNs. User

tags stand for user interests in some OSNs. Iofciu et al⁴¹ use the tag feature with the BM25 and IDF methods to link accounts among Flickr, StumbleUpon, and Delicious. However, the number of tags is not large enough to represent user interests. Fortunately, the topic model⁴² can be utilized to extract social interests from user account. For instance, Nanavati et al⁴³ employ the n-gram method to identify anonymous users with user interests extracted from their OSN comments. The topic model can be used to extract user interests from UGCs. We utilize the word2vec⁴⁴ method to represent user interests with vectors and calculate user interest similarity through vector's cosine similarity. The user interest similarity of account v_i^s and v_j^t is calculated through Equation 8:

$$Sim_{Interest}(v_i^s, v_j^t) = \cos(F, F') = \frac{F \cdot F'}{\|F\| \times \|F'\|} \quad (8)$$

where, $F = \{f_1, f_2, \dots, f_m\}$ and $F' = \{f'_1, f'_2, \dots, f'_n\}$ are vectors of account interests.

4.4 | Joint learning model

In order to integrate user profile, user online time distribution, and user interest features to measure the similarity of 2 accounts, we propose the joint learning model as shown in Equation 9:

$$Sim(v_i^s, v_j^t) = a \cdot Sim_{Profile}(v_i^s, v_j^t) + b \cdot Sim_{Time}(v_i^s, v_j^t) + c \cdot Sim_{Interest}(v_i^s, v_j^t) \quad (9)$$

where $a + b + c = 1$. The weight of these features in measure similarity of accounts are different across OSNs. If user profile is unreal or has lots of missing values, a should be low. If user online time distribution feature is unique to distinguish 2 accounts, b should be high, and vice versa. Therefore, we tuned a , b , and c in Section 5.3.2 to achieve the best performance.

4.5 | Account linking algorithm

Algorithm 2 calculates user's account linking across different OSNs to solve the problem described in Equation 1. Table 2 shows the details of algorithm 2. The specific process is as follows. At first, we fabricate some user accounts to make the number of source OSNs and target OSNs equal. Then, we extract user profile, user online time distribution, and user interest features from user account's UGCs before calculating the similarity of the candidate linking pairs. After that, we build a list of candidate pairs and sort the list according to the similarity value. Then, we use the idea of the KM algorithm to solve the weighted

TABLE 2 Algorithmic description of account linking

Algorithm 2 account linking across different OSNs
Input: Source OSN $G^s(V1, E1)$, target OSN $G^t(V2, E2)$, $G = (V = (v_i^s, v_j^t), E)$, $v_i^s \in V1$, $v_j^t \in V2$.
Output: a set of inferred account pairs M , link $(v_i^s, v_j^t) \in M$.
1: Make the length of $V1$ equal to $V2$ with the fabricate account v_j ;
2: For each account pair (v_i^s, v_j^t) , extract features;
3: Calculate $\text{sim}(v_i^s, v_j^t)$; // calculate the similarity of v_i^s and v_j^t
4: Build a preference list Q (the length is k) according the similarity of unlabeled account v_i^s and v_j^t ;
5: Sort(Q); // sort the similarity scores into a preference list of the candidate linking accounts.
6: Initialize all unlabeled v_i^s in G^s and v_j^t in G^t as free;
7: $M = \emptyset$;
8: While ($V1$)
9: {
10: Initialize the node weight with $\text{sim}(v_i^s, v_j^t)$;
11: While (find no complete matching with subgraphs)
12: {utilize the Hungarian algorithm to find complete matching;
13: If (complete matching has not been found)
14: Update the node weight;}
15: $V1 = (V1 - v_i^s)$;
16: $M = M \cup \{(v_i^s, v_j^t)\}$;
17: }
18: Remove the account pair contains fabricate account v_j from M ;
19: Return M ;

bipartite matching problem. When v_i^s is not linked to any account, we initialize the node weight with $\text{sim}(v_i^s, v_j^t)$. Then, the Hungarian Algorithm⁴⁵ is utilized to find complete matchings. If no complete matching has been found, the node weight should be updated. If the complete matching has been found, user account v_i^s should be deleted from $V1$. The process will not terminate until all the unmapped user accounts have linking accounts. Finally, we can get the linking account pairs by deleting the account pair contained fabricate accounts.

However, if the number of the user accounts is n , we need to compare n^2 similarities of account pairs. Because profile similarity is an important feature to link account pairs, we propose to use profile similarity threshold to pre-prune candidate user account pairs. User account pairs is deleted from the candidate account pairs if the profile similarity is less than the threshold. Thus, we can reduce the amount of calculating candidate user account pairs.

5 | PERFORMANCE EVALUATION

In this section, we first present datasets in Section 5.1. We then describe the compared method in Section

5.2. At last, we present our main evaluation results in Section 5.3.

5.1 | Datasets

We crawled data from Sina Weibo,^{*} Tencent Weibo,[†] Douban,[‡] and Dianping.[§] The datasets cover more than 1 million users. The users have at least 2 accounts. Among them, 42% users have at least 3 accounts. Because we crawled the data from social networks with DFS method, the data contains profile information, tweet, retweet, and comments. Therefore, the amount of information in the crawled subsets of the OSN can represent the average account in these OSNs. According to the small world theory,⁴⁶ the longest distance between 2 persons is less than 6 hops in the social network. Thus, we crawled the data following DFS

*www.weibo.com

†www.t.qq.com

‡www.douban.com

§www.dianping.com

method and set the search depth as 6. The dataset statistics is described in Table 3.

5.1.1 | Sina Weibo

Sina Weibo is a widely used microblogging service in China. The dataset used in our experiment was crawled from its website from January to June 2014, which contains 1 260 752 users and 46 083 383 friend relationships. Taking user profiles as network nodes, the comments, likes, and retweet behaviors are used to reconstruct the social network structure between nodes.

5.1.2 | Tencent Weibo

Tencent Weibo is another popular microblogging service in China, which is similar to Sina Weibo. The dataset used in our experiment was crawled from its website from January to June 2014, which contains 1 101 324 users and 27 693 893 friend relationships.

5.1.3 | Douban

Douban is a social network service for people to share comments on movies, books, music, and some off-line events in Chinese cities. The dataset was crawled from its website from January to June 2014, which contains 1 105 492 users and 34 387 876 friend relationships.

5.1.4 | Dianping

Dianping is a Chinese review site, which offers product or service reviews for local businesses such as restaurant, hotels and cinemas, booking, and group-buying services. The dataset used in our experiment was crawled from its website from January to June 2014, which contains 1 007 136 users and 29 571 930 friend relationships.

5.1.5 | Ground truth

We link user accounts in different OSNs according to the social user accounts contained in user profile. These accounts can be used as the ground truth, and these accounts are used as seed to crawl the other related accounts. The statistics of the ground truth is described in Table 4.

5.2 | Compared methods

We compare the proposed method to the existing state-of-the-art methods for account linking across OSNs.

5.2.1 | MOBIUS¹⁰

Zafarani et al model user behaviors to identify users across OSNs. They employ a supervised learning method to link the corresponding users across OSNs. Individual behavioral patterns are categorized into human limitations, exogenous factors, and endogenous factors, which are used to link user accounts across OSNs. MOBIUS utilizes l_1 -Regularized Logistic Regression to achieve the best performance. Therefore, in our experiments, we also employ l_1 -Regularized Logistic Regression as the method of choice.

5.2.2 | HYDRA²¹

Based on user profile attributes, UGC, and location trajectories, this method proposes a multi-objective optimization framework to learn the link function and the account linking function by minimizing the objective function through a unified multi-objective optimization framework. In our experiments, we employ HYDRA-M method to link user accounts, because HYDRA-M method achieves the best performance in Liu et al.²¹ We tune the parameter g_L , g_M , p , s_S , and s_D to make HYDRA-M method achieve the best performance.

5.2.3 | COSNET²⁸

Both local and global consistency are employed in COSNET to link accounts among multiple OSNs. Zhang et al use username uniqueness, profile, ego network, and social status to link user accounts, and they propose a subgradient algorithm to train the model and develop an energy-based objective function to balance the importance of these features. They use a threshold to determine whether 2 usernames belong to the same user. In our experiments, we employ the code offered by Zhang et al and set the threshold as 0.8, which is same as parameter setting in Zhang et al.²⁸ The code we used in the experiments can be found in Aminer.*

5.2.4 | NS²⁹

Narayanan and Shmatikov design a re-identification algorithm targeting at OSNs, which only uses network topology to link accounts between Twitter and Flickr. In our experiments, we use the same parameter setting as.²⁹

*<http://aminer.org/cosnet>

TABLE 3 Statistics of datasets

Network	Users	Relationships
Sina Weibo	1 260 752	46 083 383
Tencent Weibo	1 101 324	27 693 893
Douban	1 105 492	34 387 876
Dianping	1 007 136	29 571 930

TABLE 4 Statistics of the ground truth

OSN	OSN	Linking Account Pair
Tencent Weibo	Dianping	392 783
Dianping	Sina Weibo	488 667
Sina Weibo	Douban	413 675
Douban	Tencent Weibo	376 528
Sina Weibo	Tencent Weibo	432 846
Dianping	Douban	375 654

5.3 | Experimental results

We implemented our algorithm in C++ language. The proposed algorithm is conducted on a cluster with Intel Xeon E5-2620 V3 CPU, NVIDIA Tesla K80 GPU, Intel Xeon Phi 7120P, 128 GB memory, 1T SSD disk, 6T SAS disk, and CentOS release 6.4.

5.3.1 | Performance of FastDTW and Euclidean distance method

We used the Sina Weibo and Tencent Weibo datasets to evaluate the performance of FastDTW and measure user online time distribution similarity. We extract the time of tweets and measure the similarity of time sequence with the Euclidean distance and the FastDTW distance respectively. Then, we determine the similarity of user accounts according to the similarity of time sequence. When using the Euclidean distance measurement, the number of tweets per day represents user activity so as to ensure the same length of time sequence. When measured with FastDTW, the time sequence is set according to the release time of each tweet. We give the results of accuracy and efficiency in Figure 2. Figure 2A shows the account linking precision with the Euclidean distance method and the FastDTW method, respectively. As can be seen from the figure, the precision of FastDTW is 14% higher than the Euclidean distance measurement method on average. Figure 2B shows the runtime of the Euclidean distance method and the FastDTW method in measuring the similarity of the active time distribution. When the data size increases, the runtime of the

Euclidean distance method increases rapidly while the runtime of the FastDTW method grows slowly. Therefore, the FastDTW method is more efficient than the Euclidean distance method.

5.3.2 | Parameter sensitivity

After sorting the candidate matching accounts, we select k candidate matching accounts for bipartite graph matching. How can we determine the size of k to balance the accuracy and running time of the algorithm? Figure 3 shows the experimental results, where k is increased from 1 to 10 by 1. It can be seen that with the increase of k , the accuracy rate increases, so does the runtime. When k is 7, the precision achieves 94% and keeps steady. Therefore, we set k equals 7 in the experiment. Parameter k is tuned according to the real dataset, and k maybe different in different social networks.

In order to find out the values of a , b , and c to make the joint learning model achieve the best performance, we tuned a and b increased from 0.0 to 1.0 by 0.1, where $c = 1 - a - b$. We utilized Tencent Weibo & Dianping dataset to train the joint learning model. The dataset is partitioned into mapped pair set and unmapped pair set, which is used 10-fold cross validation, where 9-folds are used as the training set and 1-fold is used as the test set. Figure 4 shows 66 experimental results. We find that when $a = 0.5$, $b = 0.2$, and $c = 0.3$, the accuracy of our method is 95.19%. Thus, Table 5 shows the results of our method in comparison with other methods when $a = 0.5$, $b = 0.2$, and $c = 0.3$.

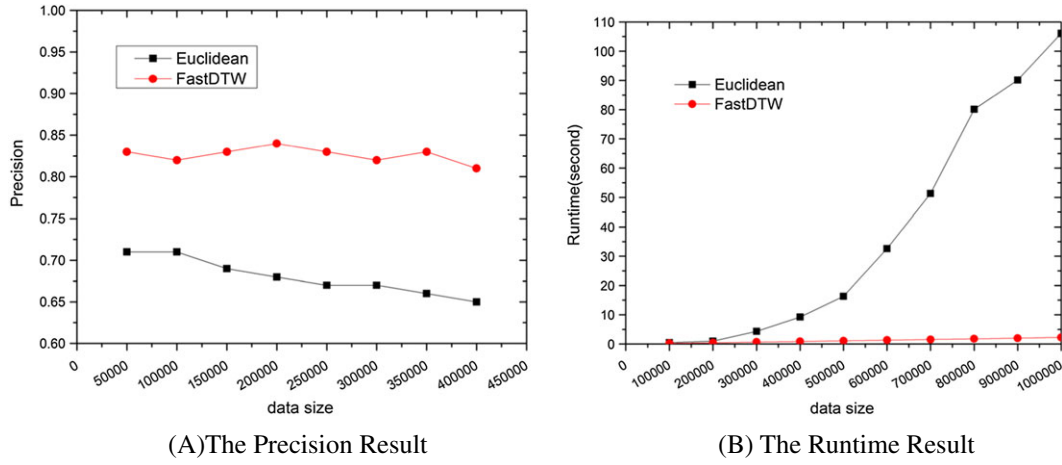


FIGURE 2 The comparison between the Euclidean distance method and the FastDTW in terms of precision and runtime results

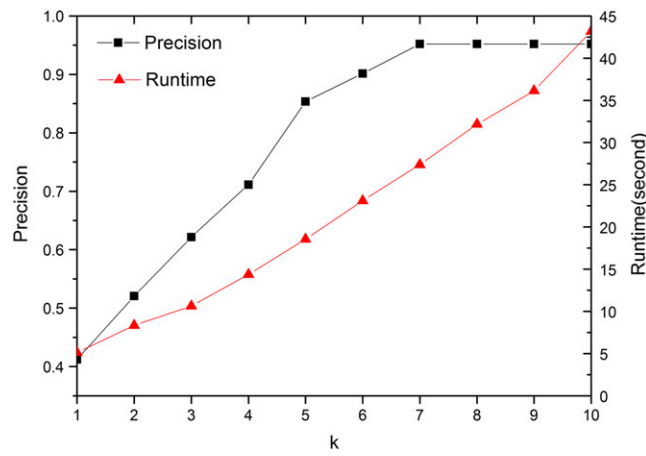


FIGURE 3 Precision and runtime comparison based on the candidate account k

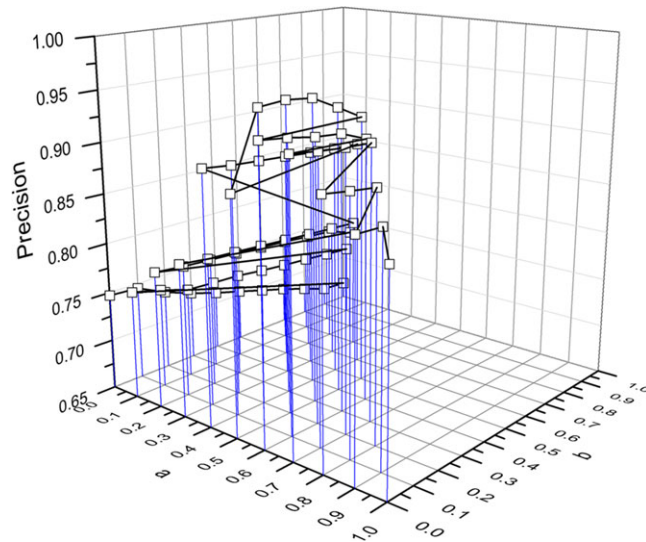


FIGURE 4 Precision with different balance factors a and b

TABLE 5 The precision recall and F1 values from different algorithms when $a = 0.5$, $b = 0.2$, and $c = 0.3$, the bold data indicate the competency performance among methods

Network pair	NS			MOBIUS			COSNET			HYDRA			Our Method		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Tencent Weibo & Dianping	83.74	58.57	68.93	60.40	73.08	66.14	78.92	71.65	75.11	90.80	53.60	67.41	89.33	76.99	82.70
Dianping & Sina Weibo	74.17	41.74	53.42	57.89	73.82	64.89	76.51	67.94	71.97	86.44	40.26	54.93	90.38	73.20	80.89
Sina Weibo & Douban	87.62	42.38	57.13	44.01	61.66	51.36	75.22	63.41	68.81	89.66	44.35	59.35	85.56	74.18	79.46
Douban & Tencent Weibo	94.96	44.83	60.91	78.38	61.29	68.79	84.20	63.37	72.31	94.79	47.11	62.94	95.19	70.92	81.28
Overall	85.12	46.88	60.10	60.17	67.46	62.79	78.71	66.59	72.05	90.42	46.33	61.16	90.12	73.82	81.08

5.3.3 | Comparison to the state-of-the-art algorithms

Table 5 shows the precision, recall, and F1 score of the compared methods. In the experiments, 90% of the dataset was used for training, and the rest 10% was used for testing. Besides that, 10-fold cross-validation was used, and the average accuracy for correctly predicting linked user accounts in the testing was recorded.

Our method achieves the best recall rates and F1 scores compared with NS, MOBIUS, COSNET, and HYDRA on the Tencent Weibo & Dianping and Sina Weibo & Douban datasets. On the Dianping & Sina Weibo and Douban & Tencent Weibo datasets, our method achieves the best results. On average, our method achieves the best results in terms of precision, recall, and F1 score, which is 90.12%, 73.82%, and 81.08%, respectively. Our method is better than the 4 existing methods in precision, recall, and F1 score by 11%, 17%, and 29% on average.

5.3.4 | Runtime comparison

We compare runtime of all methods with same dataset (Tencent Weibo & Dianping dataset) input and give the

results in Figure 5. In Figure 5, we can see MOBIUS consumes the least time as it only employs user name to link user accounts; HYDRA and NS spend more time than MOBIUS. HYDRA utilizes user profiles and UGC to map user account, thus, it spends more time than MOBIUS, and NS method only utilizes network structure to match the user accounts, as the cost of comparing the similarity of the network structure is higher than only compare user names. The cost of runtime for COSNET is the highest as it employs both network structure feature and user behaviors to link accounts. As for our method, it consumes less time than COSNET and more time than other 3 methods because KM algorithm has a runtime complexity of $O(n^3)$, which is a bottleneck for large-scale social networks with billion users. Compared with the precision and recall rate, the runtime is less important in our solution. However, it is a deficiency of our proposed method. We will explore efficiency method in the future work.

5.3.5 | Discussion

The proposed method employs user profile, user online time distribution feature, and user interest feature to

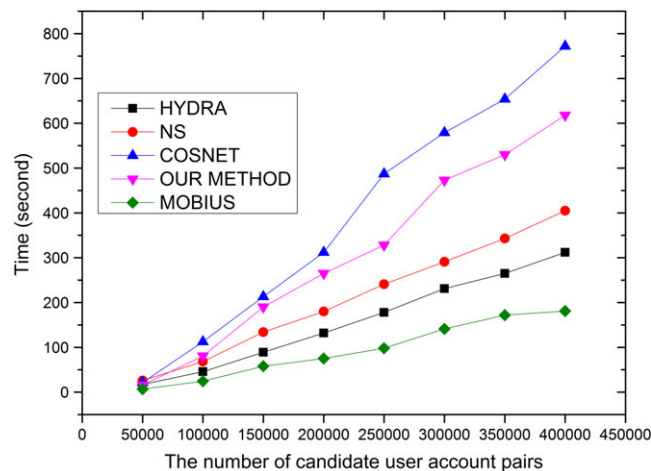


FIGURE 5 Runtime of different methods with different number of user account pairs

link accounts across social networks. Therefore, we need crawl amount of user information to extract profile feature, online time activity feature, and user interest feature. In order to reduce the cost of comparing user accounts, we utilize profile similarity threshold to discard candidate user account pairs in pre-pruning process. However, if the social network is suffering Sybil attack, and user profile information is fake, the accuracy of our method will be decreased, and the run time will be longer.

In the initial stage, we need some linked user accounts as seeds to construct the graph with DFS method based on small world theory. However, small world is not a precise theory, and some isolated user cannot be linked. Therefore, our method cannot find out all the mapped user accounts. When meeting the large-scale social networks, our method will face the bottleneck because of KM algorithm has time complexity of $O(n^3)$. This is a problem we have to solve in the future work.

6 | CONCLUSIONS

In this paper, we address the account linking problem with a maximum matching method on weighted bipartite graphs, and we develop a joint learning model to measure the similarity of user accounts. The balanced factor is used to tune the weight of user profile, user online time distribution, and user interests in measuring account similarity. We adopt the KM algorithm to solve the maximum matching on weighted bipartite graphs precisely. The experimental results on real OSNs datasets validate the effectiveness of our method. Still, there are some challenges left for our future work, for example, paralleling this algorithm for social network scalability and utilizing this method to de-anonymize user accounts in large-scale OSNs. We will put more effort to solve these challenges so as to make our method more accurate and scalable for social link inferring.

ACKNOWLEDGEMENTS

This work is supported by the National Nature Science Foundation of China (No.61402255, No.61170292, No.61373161, No.61672470), the National Key Research and Development Projects Intergovernmental Cooperation in Science and Technology of China (No.12016YFE0100600, No.12016YFE0100300), the Foundation of Henan science and Technology Department (No.162102410076, No.162102310578), Foundation of Henan Educational Committee (No.16A520062, No.17A520064), the Natural Science Foundation of Guangdong Province (No.2015A030310492), and the

Fundamental Research Project of Shenzhen Municipality (No.JCYJ20160301152145171).

ORCID

Jiangtao Ma  <http://orcid.org/0000-0001-5181-4045>

Guangwu Hu  <http://orcid.org/0000-0003-3947-9998>

REFERENCES

- Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: simple building blocks of complex networks. *Science* (80-). 2002;298(5594): 824 LP-827
- Miller G. Social scientists wade into the tweet stream. *Science* (80-). 2011;333(6051): 1814 LP-1815
- Benson AR, Gleich DF, Leskovec J. Higher-order organization of complex networks. *Science* (80-). 2016;353(6295): 163 LP-166
- (2016) The 38th China Internet Development Report.
- Goga O. (2014) Matching user accounts across online social networks: methods and applications.
- Ma H, Zhou D, Liu C, Lyu R, King I. Recommender systems with social regularization. *Proc. Fourth ACM Int. Conf. Web Search Data Min.*, 2011:287-296.
- Cui L, Sun L, Fu X, Lu N, Zhang G. Exploring {a} trust based recommendation approach for videos in online social network. *Signal Process Syst.* 2017;86(2-3):207-219.
- Han X, Sun L, Zhao J. (2011) Collective entity linking in web text: a graph-based method. *Proc. 34th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, 765-774.
- Getoor L, Machanavajjhala A. Entity resolution: theory, practice & open challenges. *Proc VLDB Endow.* 2012;5(12):2018-2019.
- Zafarani R, Tang L, Liu H. User identification across social media. *ACM Trans Knowl Discov Data.* 2015;10(2): 16:1--16:30
- Zafarani R, Liu H. Connecting corresponding identities across communities. *Int. Conf. Weblogs Soc. Media, Icwsm 2009, San Jose, California, Usa, May.* (2009).
- Perito D, Castelluccia C, Kaafar MA, Manils P. How unique and traceable are usernames? In: Fischer-Hübner S, Hopper N, eds. *Privacy Enhancing Technologies: 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg; 2011:1-17.
- Irani D, Webb S, Li K, Pu C. Large online social footprints—an emerging threat. *2009 Int Conf Comput Sci Eng.* 2009;3:271-276.
- Motoyama M, Varghese G. I seek you: searching and matching individuals in social networks. *Proc. Elev. Int. Work. Web Inf. Data Manag.*, 2009:67-75.
- Malhotra A, Totti L, Meira W, Kumaraguru P, Almeida V. Studying user footprints in different online social networks. *2012 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Min.*, 2012:1065-;1070.
- Zafarani R, Liu H. (2013) Connecting users across social media sites: a behavioral-modeling approach. *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2013:41-49.

17. Liu S, Wang S, Zhu F, Zhang J, Krishnan R. HYDRA: large-scale social identity linkage via heterogeneous behavior modeling. *Proc. 2014 ACM SIGMOD Int. Conf. Manag. Data*, 2014:51-62.
18. Almishari M, Tsudik G. Exploring linkability of user reviews. In: Foresti S, Yung M, Martinelli F, eds. *Computer Security—ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012:307-324.
19. Kong X, Zhang J, Yu PS. Inferring anchor links across multiple heterogeneous social networks. *Proc. 22Nd ACM Int. Conf. Inf. Knowl. Manag.*, 2013:179-188.
20. Zhang J, Kong X, Yu PS. Transferring heterogeneous links across location-based social networks. *Proc. 7th ACM Int. Conf. Web Search Data Min.*, 2014:303-312.
21. Liu S, Wang S, Zhu F. Structured learning from heterogeneous behavior for social identity linkage. *IEEE Trans Knowl Data Eng.* 2015;27(7):2005-2019.
22. Tan S, Guan Z, Cai D, Qin X, Bu J, Chen C. Mapping users across networks by manifold alignment on hypergraph. *28th AAAI Conf. Artif. Intell.*, 2014:159-165.
23. Cui Y, Pei J, Tang G, Luk W, Jiang D, Hua M. Finding email correspondents in online social networks. *World Wide Web*. 2013;16(2):195-218.
24. Bartunov S, Korshunov A, Park S, Ryu W, Lee H. Joint link-attribute user identity resolution in online social networks. *Proc. 6th Int. Conf. Knowl. Discov. Data Mining, Work. Soc. Netw. Min. Anal. ACM*. 2012.
25. Koutra D, Tong H, Lubensky D. Big-align: fast bipartite graph alignment. *2013 IEEE 13th Int. Conf. Data Min.*, 2013:389-398.
26. Zhou X, Liang X, Zhang H, Ma Y. Cross-platform identification of anonymous identical users in multiple social media networks. *IEEE Trans Knowl Data Eng.* 2016;24(9):411-424.
27. Zhang J, Chen J, Zhu J, Chang Y, Yu P. Link prediction with cardinality constraint. *Proc. Tenth ACM Int. Conf. Web Search Data Min.*, 2017:121-130.
28. Zhang Y, Tang J, Yang Z, Pei J, Yu P. COSNET: connecting heterogeneous social networks with local and global consistency. *ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2015:1485-1494.
29. Narayanan A, Shmatikov V. De-anonymizing social networks. *2009 30th IEEE Symp. Secur. Priv.*, 2009:173-187.
30. Narayanan A, Shmatikov V. Robust de-anonymization of large sparse datasets. *2008 IEEE Symp. Secur. Priv. (sp 2008)*, 2008:111-125.
31. Narayanan A, Shi E, Rubinstein BIP. Link prediction by de-anonymization: how we won the Kaggle social network challenge. *Int Jt Conf Neural Netw.* 2011;1825-1834.
32. Sharad K, Danezis G. An automated social graph de-anonymization technique. *Proc. 13th Work. Priv. Electron. Soc.*, 2014:47-58.
33. Korula N, Lattanzi S. An efficient reconciliation algorithm for social networks. *Proc VLDB Endow.* 2014;7(5):377-388.
34. Backstrom L, Dwork C, Kleinberg J. Wherefore art thou R3579x?: anonymized social networks, hidden patterns, and structural steganography. *Proc. 16th Int. Conf. World Wide Web*, 2007:181-190.
35. Zhou B, Pei J. Preserving privacy in social networks against neighborhood attacks. *2008 IEEE 24th Int. Conf. Data Eng.*, 2008:506-515.
36. Zhou B, Pei J. The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowl Inf Syst.* 2011;28(1):47-77.
37. Wondracek G, Holz T, Kirde E, Kruegel C. A practical attack to de-anonymize social network users. *Secur Priv.* 2010;223-238.
38. Qian J, Li XY, Zhang C, Chen L. De-anonymizing social networks and inferring private attributes using knowledge graphs. *IEEE INFOCOM 2016 - 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016:1-9.
39. Ji S, Li W, Gong NZ, Mittal P, Beyah R. Seed-based de-anonymizability quantification of social networks. *IEEE Trans Inf Forensics Secur.* 2016;11(7):1398-1411.
40. Salvador S, Chan P. Toward accurate dynamic time warping in linear time and space. *Intell Data Anal.* 2007;11(5):561-580.
41. Iofciu T, Fankhauser P, Abel F, Bischoff K. Identifying users across social tagging systems. *Int. Conf. Weblogs Soc. Media, Barcelona, Catalonia, Spain, July*. 2010.
42. Ramage D, Hall D, Nallapati R, Manning CD. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. *Proc. 2009 Conf. Empir. Methods Nat. Lang. Process. Vol. 1-Volume 1*, 2009:248-256.
43. Nanavati M, Taylor N, Aiello W, Warfield A. Herbert West-Deanonymizer. *HotSec*. 2011.
44. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Burges CJC, Bottou L, Welling M, et al., eds. *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc.; 2013:3111-3119.
45. Bellur U, Kulkarni R. Improved matchmaking algorithm for semantic web services based on bipartite graph matching. *Web Serv. 2007. ICWS 2007. IEEE Int. Conf.*, 2007:86-93.
46. Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. *Nature*. 1998;393(6684):440-442.

How to cite this article: Ma J, Qiao Y, Hu G, et al. Social account linking via weighted bipartite graph matching. *Int J Commun Syst.* 2018;31:e3471. <https://doi.org/10.1002/dac.3471>