Sun *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:211
https://doi.org/10.1186/s13638-019-1538-6

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Privacy-aware friend finding in social network based on thumbs-up data

Zewei Sun[1], Huaizhen Kou[2] and Wanli Huang[2*]

Abstract

Social network has provided a promising way for massive users to share their ideas and communicate with each other. A key issue in social network is to find out the prospective friends of users so as to extend the users' social cycles. Fortunately, users' thumbs-up data on web news or blogs have become an important evaluation basis in friend finding. Typically, through analyzing the thumbs-up data from different users, we can find out the friends or neighbors of a user. However, the thumbs-up data are often sensitive to users as they can disclose the private information of users, which violate the civil privacy-protection laws enacted by governments. In view of this challenge, we introduce the Simhash technique in information retrieval domain into social network and further bring forth a privacy-aware prospective friend-finding solution in social network based on the sensitive thumbs-up data. At last, we conduct a range of experiments based on well-known *Movielens* dataset. Experimental data demonstrate the advantages of our solution.

Keywords: Social network, Friend finding, Privacy, Thumbs-up data, Simhash

1 Introduction

The ever-increasing maturity of web techniques [1–5] has promoted the birth of various social networks, such as Wechat, Facebook, and Twitter. Generally, in a social network, a user member can communicate with other friends or neighbors, share his or her ideas, and make other interesting collaborations [6–8]. Today, an increasing number of users have joined various social networks and become a part of the whole web ecosystem [9, 10].

At the very beginning, a user's social network is often very small as he or she only has a few neighbors or friends. Such a small social network does not make much sense for the user as it cannot deeply fulfill the sharing objective of social software. Therefore, in the domain of social network, a key issue that needs to be addressed is how to find out the prospective friends or neighbors of a user so as to effectively extend the user' friend network [11, 12]. In other words, making new friends is a practical and significant requirement for the users in social network, which is also one of the major reasons why users are willing to use the social interaction software or tools.

Fortunately, the thumbs-up data left by users have provided us a promising way to seek for the prospective friends or neighbors of a user in social network. Generally, a user will leave a thumbs-up record towards a shared article in social network if he or she likes the article. Therefore, the thumbs-up data from a user are a good basis to evaluate the preferences of the user. Furthermore, if the thumbs-up data from two users A and B are close, then we can infer that A and B are possible friends or neighbors as they share similar preferences. Therefore, through analyzing the users' thumbs-up data, we can quickly figure out the users who may become friends or neighbors of a target user in the future.

However, the thumbs-up data left by users are a kind of sensitive data as they probably disclose the private information of users [13–16]. Such a kind of privacy disclosure often does harm to the information sharing in social network and violates the correlated civil privacy-protection laws enacted by government departments. In addition, the thumbs-up data-based friend-finding process is often time-consuming as a social network often contains massive users, and each user often left multiple thumbs-up data.

* Correspondence: wanlih1983@126.com
[2]School of Information Science and Engineering, Qufu Normal University, Rizhao, China
Full list of author information is available at the end of the article

In view of the above two challenges, Simhash technique that is developed for quick information retrieval is introduced for friend or neighbor finding in social network. Afterwards, a privacy-aware and time-efficient friend-finding approach is brought forth based on Simhash.

In summary, the scientific contributions of our paper are threefold:

(1) We introduce the Simhash technique in information retrieval domain into social network to help find the prospective friends or neighbors of a user.
(2) We develop a Simhash-based friend- or neighbor-finding approach that can protect the sensitive information contained in thumbs-up data.
(3) A series of experiments are conducted on well-known *Movielens* dataset. The reported experimental results show the feasibility of our proposed solution.

The rest of paper is organized as follows. In Section 2, we formulate the thumbs-up data-based friend- or neighbor-finding problem in social network. Our proposal based on Simhash is presented in Section 3. In Section 4, we conduct a set of experiments to validate the feasibility of our solution; correlated comparison analyses are also presented. At last, in Section 5, we conclude our paper and point out the prospective research directions in the future.

2 Formulation

To better clarify the friend- or neighbor-finding problem in social network, we list the formal symbols to be used in this paper as below:

(1) *MEMBER* = $\{M_1, \ldots, M_m\}$: user members in a social network.
(2) *ARTICLE* = $\{A_1, \ldots, A_n\}$: articles shared in a social network.
(3) $u^\#$: a user that needs to find new friends or neighbors.
(4) $\text{THUMBS-UP} = (T_1, \ldots, T_m) = \begin{bmatrix} t_{1,1} & \ldots & t_{1,n} \\ \vdots & \ddots & \vdots \\ t_{m,1} & \cdots & t_{m,n} \end{bmatrix}$

Thumbs-up m^*n matrix left by users: each row denotes a member in social network and each column represents an article shared in the social network. Here, $t_{i,\ j} = 1$ if member M_i leaves a thumbs-up record over article A_j; $t_{i,\ j} = 0$ if member M_i does not leave a thumbs-up record over article A_j. In other words, $t_{i,\ j}$ is a Boolean value.

With abovementioned formulations, we describe the friend- or neighbor-finding process in social network as below: according to the thumbs-up data in *THUMBS-UP* matrix associated with articles in set ARTICLE and users in set MEMBER, find out the prospective neighbors or friends of user $u^\#$. As the thumbs-up data are sometimes sensitive to users, we need to protect the private information contained in *THUMBS-UP* matrix. This is the main issue that needs to be addressed in this paper.

Next, we utilize the example in Fig. 1 to motivate our paper. In the figure, there are three user members in the social network, i.e., M_1, M_2, and M_3, and four articles, i.e., A_1, A_2, A_3, and A_4. Each arrow between members and articles in the figure denotes that there is a thumbs-up record over an article by a member. Concretely, M_1 left two thumbs-up records on articles A_1 and A_3, respectively; M_2 left two thumbs-up records on articles A_2 and A_3, respectively; M_3 left two thumbs-up records on articles A_1 and A_4, respectively; and $u^\#$ left three thumbs-up records on articles A_1, A_2 and A_3, respectively.

Generally, through analyzing and comparing the thumbs-up data left by three user members and $u^\#$, we can find out the friends or neighbors that share similar or same preferences with user $u^\#$, so as to finish the friend or neighbor recommendation process. However, the thumbs-up data are often sensitive to the three members and $u^\#$; therefore, they often refuse to release their thumbs-up data to other parties due to the probable privacy disclosure risks. Therefore, it becomes a challenging task to find the friends or neighbors of $u^\#$ without revealing the real thumbs-up data. This is the major focus of this paper, and we will introduce how to achieve the above objective based on well-known Simhash technique. Detailed introduction will be presented in the next section.

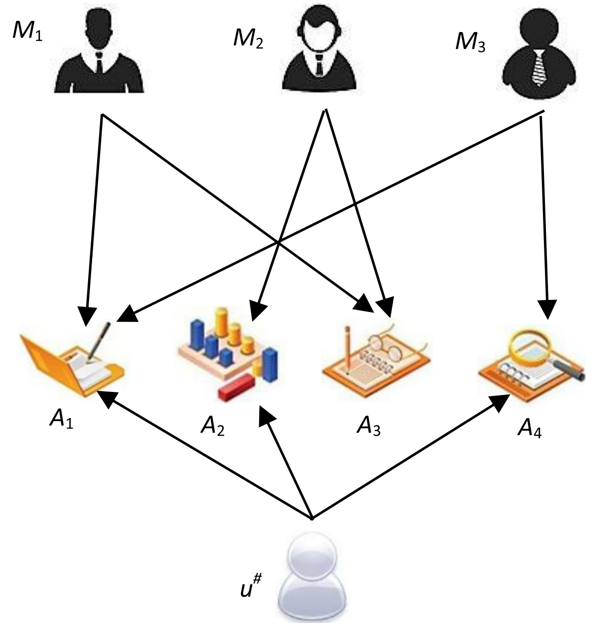


Fig. 1 Privacy-aware friend-finding based on thumbs-up data: an example

3 Simhash-based friend finding with privacy preservation

Next, we will introduce our proposed friend- or neighbor-finding approach (named $\text{FF}_{\text{Simhash}}$) based on *Simhash* and sensitive thumbs-up data. The privacy-preservation strategy of our solution is as follows: first, we generate user member indices offline based on thumbs-up data and Simhash technique; second, we search for similar friends or neighbors of $u^{\#}$ based on the derived member indices; finally, the selected Top-*N* friends are returned to $u^{\#}$.

Concretely, the proposed $\text{FF}_{\text{Simhash}}$ solution mainly consists of the three steps in Fig. 2. Here, *THUMBS-UP* denotes the thumbs-up data set left by user members in social network; $u^{\#}$ is a user who needs to find new friends or neighbors; Top-*N* (here, *N* is a parameter that should be designated by $u^{\#}$) means the size of returned neighbor set or friend set requested by $u^{\#}$; for example, if $N = 3$, then the optimal three friends or neighbors would be returned to $u^{\#}$.

3.1 Step 1: Generate user member indices offline based on Simhash and thumbs-up data

The THUMBS-UP set introduced in Section 2 records the historical thumbs-up behaviors of all the user members in a social network. If a user member likes an article, then a thumbs-up action is triggered; otherwise, not. Therefore, the thumbs-up data in *THUMBS-UP* can be regarded as an ideal basis to evaluate the personalized preferences of a user member. Inspired by this observation, we can extract hidden user preferences by analyzing the thumbs-up behaviors of the user. In other words, we can use the thumbs-up data of a user to delegate his (or her) preferences and index.

Concretely, we utilize Simhash technique [17] to generate user member indices based on the thumbs-up data in *THUMBS-UP* set. Next, we introduce the concrete indices generation process based on Simhash. In set *ARTICLE*, there are totally *n* articles $A_1, \ldots, A_n$. So we can encode these *n* articles with $\lceil \log_2^n \rceil$ 0/1 bits (here, $\lceil x \rceil$represents the upper bound of value *x*). For example, we can encode 3 articles A_1, A_2, A_3 with 2 ($\lceil \log_2^3 \rceil = 2$) Boolean bits:

$$A_1 = (0\ 1), A_2 = (1\ 0), A_3 = (1\ 1)$$

This way, we can convert the *n* articles $A_1, \ldots, A_n$ into corresponding 0/1 expressions. Next, in the 0/1 expressions, the "0" values are replaced by "− 1"; for example, $A_1 = (-1\ 1)$, $A_2 = (1\ -1)$, $A_3 = (1\ 1)$.

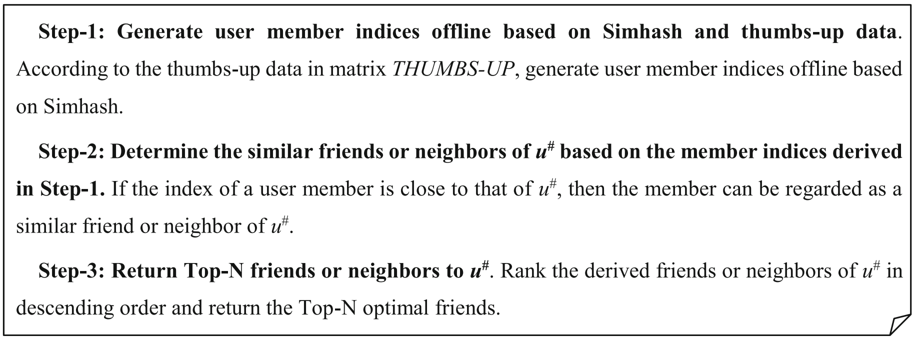
In *THUMBS-UP*, each row denotes a user member; for example, the *i*-th row $T_i = (t_{i,1}, \ldots, t_{i,n})$. Here, $t_{i,j} = 1$ if user member M_i leaves a thumbs-up record over article A_j; $t_{i,j} = 0$ if user member M_i does not leave a thumbs-up record over article A_j. Thus for user member M_i ($1 \le i \le m$), we multiply A_j and $t_{i,j}$, respectively ($1 \le j \le n$). Let us consider the above example with three articles $A_1 = (-1\ 1)$, $A_2 = (1\ -1)$, $A_3 = (1\ 1)$. If M_i only leaves thumbs-up records on A_1 and A_3, i.e., $T_i = (1, 0, 1)$, then we can get the following multiplication operations:

$$A_1 * 1 = (-1\ 1)$$

$$A_2 * 0 = (0\ 0)$$

$$A_3 * 1 = (1\ 1)$$

Next, we calculate the sum of each column of the above equations, i.e., $A_1 * 1 + A_2 * 0 + A_3 * 1 = (0, 2)$. Then, the negative values are replaced by "0", and positive values are replaced by "1". Thus, we can get $A_1 * 1 + A_2 * 0 + A_3 * 1 = (0, 1)$. According to the Simhash theory, "01" can be considered as the index for user member M_i based on the thumbs-up data in set *THUMBS-UP*. Compared to the original thumbs-up data left by M_i, i.e., $T_i = (1, 0, 1)$, index "01" contains less user privacy; this is also one reason that we choose the Simhash technique to perform privacy-aware friend or neighbor findings in social network. For each user member in the social network, we can repeat the above



Step-1: Generate user member indices offline based on Simhash and thumbs-up data. According to the thumbs-up data in matrix *THUMBS-UP*, generate user member indices offline based on Simhash.

Step-2: Determine the similar friends or neighbors of $u^{\#}$ based on the member indices derived in Step-1. If the index of a user member is close to that of $u^{\#}$, then the member can be regarded as a similar friend or neighbor of $u^{\#}$.

Step-3: Return Top-N friends or neighbors to $u^{\#}$. Rank the derived friends or neighbors of $u^{\#}$ in descending order and return the Top-N optimal friends.

Fig. 2 Concrete process of $FF_{Simhash}$

process to generate his or her less-sensitive index, respectively.

3.2 Step 2: Determine the similar friends or neighbors of $u^{\#}$ based on the member indices derived in step 1

In step 1, each user member M_i is assigned a $\lceil \log_2^n \rceil$-dimensional index, denoted by $h(M_i)$. According to the Simhash theory, if the index values of two user members are close enough, then we can come to a conclusion that these two members are friends or neighbors. Concretely, consider two members M_a and M_b whose index values are $h(M_a)$ and $h(M_b)$, respectively. If $h(M_a) \oplus h(M_b) \leq 3$, then it can be deemed that M_a and M_b are similar with large probability. Here, "$\oplus$" denotes the XOR operation that can count the difference between two multi-dimensional vectors; for example, $11111 \oplus 10111 = 1$. This way, we can judge whether a user member M_i is a qualified friend or neighbor of $u^{\#}$ by evaluating the condition $h(M_i) \oplus h(u^{\#}) \leq 3$. If M_i is a qualified friend of $u^{\#}$, then we put M_i into set *Friend* ($u^{\#}$).

3.3 Step 3: Return Top-*N* friends or neighbors to $u^{\#}$

In step 2, we have obtained a set of friends or neighbors of $u^{\#}$, i.e., *Friend* ($u^{\#}$). If the size of Friend ($u^{\#}$) is smaller than the number (i.e., N) of requested friend by $u^{\#}$, i.e., | Friend ($u^{\#}$) | $\leq N$, then all the user members in set Friend ($u^{\#}$) are returned to $u^{\#}$ and the friend-finding process for $u^{\#}$ ends successfully. Otherwise, if | Friend ($u^{\#}$) | $> N$, then we need to rank the members in set Friend ($u^{\#}$) and finally pick out N optimal ones. Here, we adopt the following simple comparison operation to evaluate the members in set Friend ($u^{\#}$). Concretely, all the members M_i ($1 \leq i \leq m$) in Friend ($u^{\#}$) can be divided into the following four cases:

$$h(M_i) \oplus h\left(u^{\#}\right) = 0 \tag{1}$$

$$h(M_i) \oplus h\left(u^{\#}\right) = 1 \tag{2}$$

$$h(M_i) \oplus h\left(u^{\#}\right) = 2 \tag{3}$$

$$h(M_i) \oplus h\left(u^{\#}\right) = 3 \tag{4}$$

We argue that the members belonging to case (3) are more similar than those belonging to case (4); likewise, the members belonging to case (2) are more similar than those belonging to case (3), and the members belonging to case (1) are more similar than those belonging to case (2). Therefore, when picking out N optimal friends or neighbors of $u^{\#}$ from set Friend ($u^{\#}$), we first consider the members belonging to case (1), then case (2), case (3), and case (4). Furthermore, if the volume of members belonging to case (1) is larger than N, then we randomly select N members from the candidates. Finally, the picked Top-N friends or neighbors of $u^{\#}$ are returned to $u^{\#}$.

This is the end of our proposed privacy-aware friend-finding approach $\mathrm{FF}_{\mathrm{Simhash}}$. To validate the feasibility of our approach, a series of experiments on a real-world dataset are designed, which will be introduced in detail in the next section.

4 Experiments

4.1 Experimental settings

In this section, a series of experiments are designed to prove the effectiveness and efficiency of our suggested $\mathrm{FF}_{\mathrm{Simhash}}$ approach in handling the privacy-aware friend finding in social network. The experiments are based on the popular *Movielens* [18] dataset containing the user-movie watching records of 4021 users and 1571 movies. To show the advantages of our proposal, we compare it with three state-of-the-art ones that are introduced briefly as below one by one:

(1) Random: this approach randomly selects a user member from the social network as the prospective friends or neighbors of user $u^{\#}$.
(2) *Core-user* [19]: this approach first searches for the core users (i.e., key users) with maximal social influences from user candidates and then finds out the friends of user $u^{\#}$ based on the core users.
(3) *Exact-match*: this approach exactly compares the feedback records left by user members and $u^{\#}$; if their feedback records are exactly the same, then they can be regarded as similar friends or neighbors.

Concretely, we compare the four approaches from the following two perspectives.

(1) Number of friends: For the friend-finding approaches in social network, the number of returned friends of $u^{\#}$ is a key factor to evaluate the performance of different approaches. For a friend-finding approach, we always expect that the returned friends of $u^{\#}$ to be suitable enough (i.e., neither too many nor too few).
(2) Time cost: the time consumed to find out the final similar friends or neighbors of $u^{\#}$. For a friend-finding approach, we expect the time cost to be small enough so as to support the real-time social interactions.

The experiments were running on a laptop with an i7 processor (2.50GHz), 16.0 GB RAM, Windows 10 OS, and Python 3.7. To alleviate the unexpected influences brought by computer network or other environment factors, each test was repeated 100 times totally and finally we report the average results.

4.2 Experimental results

Concretely, we run the following three tests and comparisons, respectively. Correlated parameter specifications can be found in Section 2.

4.2.1 Profile 1: Number of friends of $u^{\#}$

The number of returned friends or neighbors of $u^{\#}$ is a good criterion to evaluate the performances of a friend search approach in social networks. Here, for $\mathrm{FF}_{\mathrm{Simhash}}$ approach, we focus on the number of returned friends or neighbors of $u^{\#}$ in step 2 and step 3. More specifically, we list four filtering conditions for friend or neighbor search in step 3. According to Simhash theory, different search conditions in $\mathrm{FF}_{\mathrm{Simhash}}$ approach will lead to different number of returned friends or neighbors of $u^{\#}$.

We test and compare the number of friends of $u^{\#}$ of four approaches with respect to different parameters of m (varied from 50 to 900) and n (varied from 50 to 1571). Here, the number of expected friends or neighbors, i.e., $N = 3$. Running results are presented in Fig. 3. Concretely, in Fig. 3a, $n = 1571$, while in Fig. 3b, $m = 4021$.

As the figure shows, the number of users of $\mathrm{FF}_{\mathrm{Simhash}}$ approach increases with the growth of both m and n. This is because when the user volume or movie volume increases, the probability of finding a qualified friend of $u^{\#}$ also rises. Besides, our solution can always return a set of friends for $u^{\#}$ regardless of the user volume and movie volume in the social network. So the experimental results indicate that our solution can achieve a good balance between friend-finding accuracy and privacy preservation. As to the Random method and Core-user method, their returned friend number is very small; therefore, related data are not presented in Fig. 3b.

4.2.2 Profile 2: Time cost comparisons of four approaches

Time cost is another objective criterion to evaluate the friend-finding performances in social network. For a friend-finding approach, we expect the time cost to be

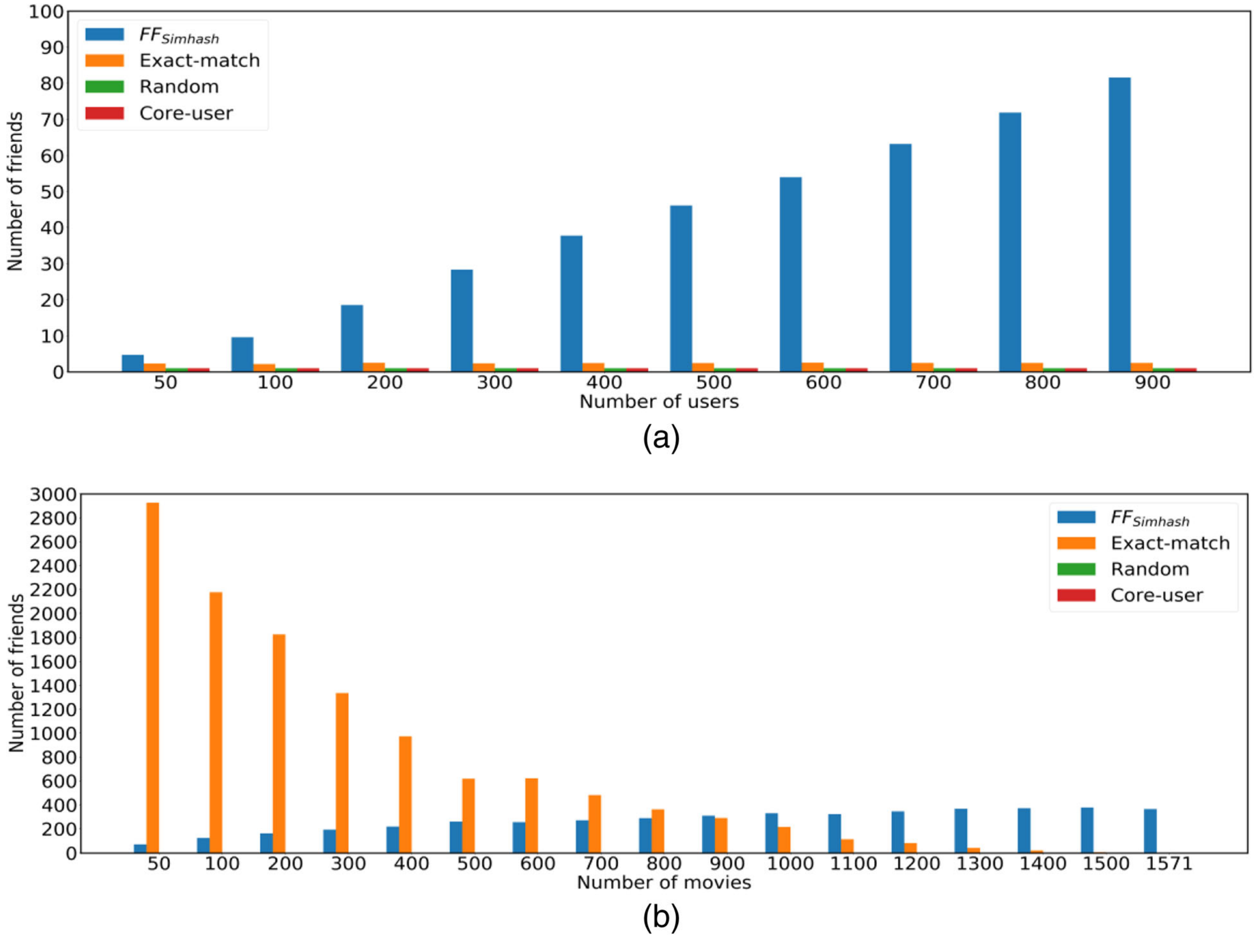


Fig. 3 **a**, **b** Number of returned friends of $u^{\#}$

small enough so as to support the real-time social interactions. Considering this, we test and compare the friend search efficiency of four competitive approaches. The number of user members in social network, i.e., m, is varied from 50 to 900; the number of movies in social network, i.e., n, is varied from 50 to 1571. Experimental results are demonstrated in Fig. 4. Here, we use "−log(t)" to indicate the time costs of different approaches (larger is better).

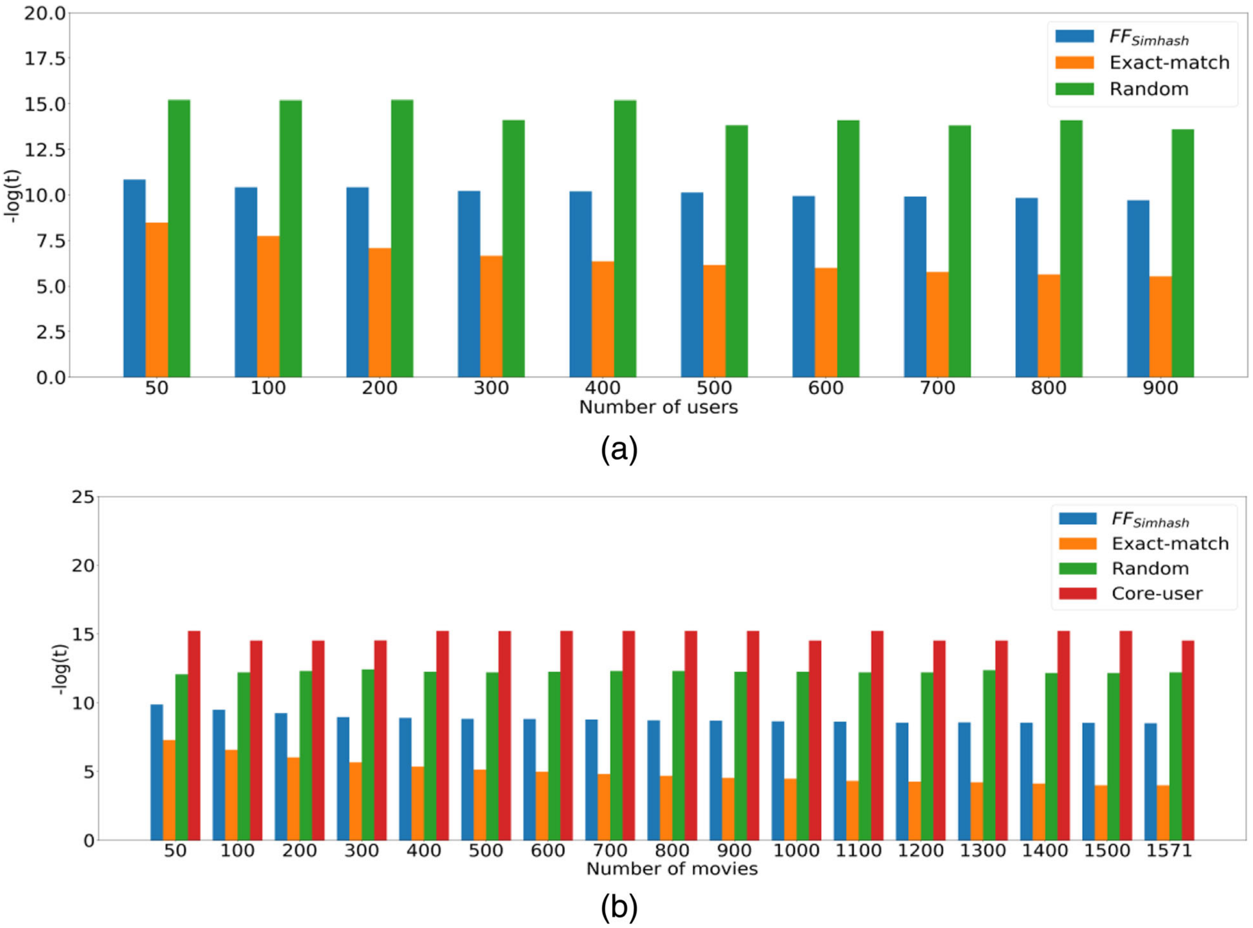
As the results show, the time cost of *Exa*ct-match approach is the highest as all the thumbs-up data should be taken into consideration when performing exact matching. Our suggested $\text{FF}_{\text{Simhash}}$ approach only requires little time to derive a set of qualified friends of $u^{\#}$ as the adopted Simhash in our solution is a time-efficient neighbor search technique, and the indices of user members in social network can be generated off-line before we perform friend or neighbor search. As to the Core-user method, the returned friend number is very small; therefore, related data are not presented in Fig. 4a.

4.3 Discussions

However, there are still several potential shortcomings in the experiments.

(1) Privacy concerns are existing in many data-related application areas [20–23]. Although we adopt Simhash technique to protect user privacy, the capability of privacy-protection effects is not measured here due to the inherent nature of Simhash technique. Besides, we only consider time cost for friend finding without considering other costs such as energy consumption [24–29] and space granularity [30, 31]. In the future, more discussions about various costs should be added.

(2) In this paper, we only discuss one dimension for friend-finding decision-makings, i.e., user members' thumbs-up data on articles in a social network. While multi-dimensional cases are more common in practical business applications [32–37]. Besides, a user may leave multiple thumbs-up records on an article, which also involves the weighting problem



(a)

(b)

Fig. 4 a, **b** Time cost with m and n

associated with the multiple thumbs-up data for an identical paper. Therefore, multiple dimensions for friend finding as well as their weighting significance are necessary to be studied in the future research work.

(3) In data-driven business applications, there is often a tradeoff between accuracy and privacy. Therefore, our proposal cannot guarantee 100% accuracy of found friends. Besides, recommendation failures are possible especially when the thumbs-up data for friend finding are very sparse, which may decrease the robustness of friend finding. In the future, it is necessary to explore further improvements in terms of accuracy and robustness while protecting the private information of user members.

5 Conclusions and future work

Social network has provided a promising way for massive users to share their ideas and communicate with each other. A key issue in social network is to find out the prospective friends of users so as to extend the users' social cycles. Typically, through analyzing the thumbs-up data from different users, we can find out the friends or neighbors of a user. However, the thumbs-up data are often sensitive to users as they can disclose the private information of users, which violate the civil privacy-protection laws enacted by governments. In view of this challenge, we introduce the Simhash technique in information retrieval domain into social network and further bring forth a privacy-aware prospective friend-finding solution in social network based on the sensitive thumbs-up data. At last, we conduct a range of experiments based on well-known Movielens dataset. Experimental data demonstrate the advantages of our solution.

In the future, we will further extend our friend-finding approach to accommodate more general and comprehensive multi-dimensional cases. Besides, how to continue to refine our work for improving the accuracy of found friends is still an open problem that needs intensive study.

Acknowledgements
Not applicable.

Authors' contributions
ZS finished the algorithm and English writing of the paper. HK finished the experiments. WH put forward the idea of this paper. All authors read and approved the final manuscript.

Funding
Partial work of this paper is supported by the Natural Science Foundation of China (No. 61872219).

Availability of data and materials
The recruited experiment dataset *Movielens* is available at http://www.grouplens.org/.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]School of Political Science and Public Administration, Qufu Normal University, Rizhao, China. [2]School of Information Science and Engineering, Qufu Normal University, Rizhao, China.

Received: 3 July 2019 Accepted: 13 August 2019
Published online: 27 August 2019

References

1. Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, X. Shen, TOFFEE: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing. IEEE Transactions on Cloud Computing, accepted (2019). https://doi.org/10.1109/TCC.2019.2923692
2. S. Wang, A. Zhou, R. Bao, C. Wu, S.S. Yau, Towards green service composition approach in the cloud. IEEE Transactions on Services Computing (2018). https://doi.org/10.1109/TSC.2018.2868356
3. Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, C. Hus, User allocation-aware edge cloud placement in mobile edge computing. Software: Practice and Experience (2019) https://doi.org/10.1002/spe.2685
4. Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, X. Shen, Energy efficient dynamic offloading in mobile edge computing for internet of things. IEEE Transactions on Cloud Computing (2019). https://doi.org/10.1109/TCC.2019.2898657
5. S. Wang, A. Zhou, M. Yang, L. Sun, C. Hsu, F. Yang, Service composition in cyber-physical-social systems. IEEE Transactions on Emerging Topics in Computing (2017). https://doi.org/10.1109/TETC.2017.2675479
6. X. Zhou, W. Liang, K. Wang, R. Huang, Q. Jin, Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data. IEEE Transactions on Emerging Topics in Computing (2018). https://doi.org/10.1109/TETC.2018.2860051.
7. L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, J. Chen, A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment. Future Generation Computer Systems **88**, 636–643 (2018)
8. W. Gong, L. Qi, Y. Xu. Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. Wireless Communications and Mobile Computing, vol. 2018, ID 3075849, 8 pages, 2018.
9. S. Wang, A. Zhou, F. Yang, R. Chang, Towards network-aware service composition in the cloud. IEEE Transactions on Cloud Computing (2016). https://doi.org/10.1109/TCC.2016.2603504
10. S. Wang, Y. Zhao, J. Xu, J. Yuan, C. Hsu, Edge server placement in mobile edge computing. Journal of Parallel and Distributed Computing **127**, 160–168 (2019)
11. X. Zhou, B. Wu, Q. Jin, Analysis of user network and correlation for community discovery based on topic-aware similarity and behavioral influence. IEEE Transactions on Human-Machine Systems **48**(6), 559–571 (2018)
12. L. Qi, W. Dou, W. Wang, G. Li, H. Yu, S. Wan, Dynamic mobile crowdsourcing selection for electricity load forecasting. IEEE Access **6**, 46926–46937 (2018)
13. S. Zhang, X. Li, Z. Tan, T. Peng, G. Wang, A caching and spatial k-anonymity driven privacy enhancement scheme in continuous location-based services. Future Generation Computer Systems. **94**, 40–50 (2019)
14. L. Qi, S. Meng, X. Zhang, R. Wang, X. Xu, Z. Zhou, W. Dou, An exception handling approach for privacy-preserving service recommendation failure in a cloud environment. Sensors **18**(7), 1–11 (2018)
15. S. Zhang, X. Mao, K.K.R. Choo, T. Peng, G. Wang, A trajectory privacy-preserving scheme based on a dual-k mechanism for continuous location-based services. Information Sciences. (2019). https://doi.org/10.1016/j.ins.2019.05.054
16. Y. Xu, L. Qi, W. Dou, J. Yu, Privacy-preserving and scalable service recommendation based on simhash in a distributed cloud environment. Complexity (2017) Article ID 3437854, 9 pages, 2017
17. S. Caitlin and G. Levin. Simhash: Hash-based similarity detection. Available: www.googlecode.com / sun / trunk / paper/Sim Hash with Bib. pdf, 2007.
18. http://www.grouplens.org/ (accessed on 24 Mar 2019).
19. G. Cao, L. Kuang, *Identifying core users based on trust relationships and interest similarity in recommender system* (IEEE International Conference on Web Services, New York, 2016), pp. 284–291

20. L. Qi, X. Zhang, W. Dou, Q. Ni, A distributed locality-sensitive hashing based approach for cloud service recommendation from multi-source data. IEEE Journal on Selected Areas in Communications **35**(11), 2616–2624 (2017)
21. S. Zhang, G. Wang, M.Z.A. Bhuiyan, Q. Liu, A dual privacy preserving scheme in continuous location-based services. IEEE Internet of Things Journal. **5**(5), 4191–4200 (2018)
22. L. Qi, R. Wang, S. Li, Q. He, X. Xu, C. Hu, Time-aware distributed service recommendation with privacy-preservation. Information Sciences **480**, 354–364 (2019)
23. S. Zhang, K.K.R. Choo, Q. Liu, G. Wang, Enhancing privacy through uniform grid and caching in location-based services. Future Generation Computer Systems **86**, 881–892 (2018)
24. X. Xu, D. Li, Z. Dai, S. Li, X. Chen, A heuristic offloading method for deep learning edge services in 5G networks. IEEE Access (2019). https://doi.org/10.1109/ACCESS.2019.2918585
25. X. Xu, R. Huang, R. Dou, Y. Li, J. Zhang, T. Huang, W. Yu, Energy-efficient cloudlet management for privacy preservation in wireless metropolitan area networks. Security and Communication Networks **2018**(8180451), 1–13 (2018)
26. Y. Chen, N. Zhang, Y. Zhang, X. Chen, Dynamic computation offloading in edge computing for internet of things. IEEE Internet of Things Journal **6**(3), 4242–4251 (2019)
27. X. Xu, S. Fu, Q. Cai, W. Tian, W. Liu, W. Dou, X. Sun, A.X. Liu, Dynamic resource allocation for load balancing in fog environment. Wireless Communications and Mobile Computing **2018**(6421607), 1–15 (2018)
28. L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, X. Xu, A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. World Wide Web Journal (2019). https://doi.org/10.1007/s11280-019-00684-y
29. X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, S. Yu, A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems. Future Generation Computer Systems (2017). https://doi.org/10.1016/j.future.2017.08.057
30. Y. Zhang, Y. Zhou, F. Wang, et al., Service recommendation based on quotient space granularity analysis and covering algorithm on Spark. Knowledge-Based Systems **147**, 25–35 (2018)
31. Y. Zhang, K. Wang, Q. He, et al., Covering-based web service quality prediction via neighborhood-aware matrix factorization. IEEE Transactions on Services Computing (2019). https://doi.org/10.1109/TSC.2019.2891517
32. L. Qi, P. Dai, J. Yu, Z. Zhou, Y. Xu, "Time-location-frequency"-aware internet of things service selection based on historical records. International Journal of Distributed Sensor Networks **13**(1), 1–9 (2017)
33. Y. Chen, J. Huang, C. Lin, X. Shen, Multi-objective service composition with QoS dependencies. IEEE Transactions on Cloud Computing **7**(2), 537–552 (2019)
34. L. Qi, J. Yu, Z. Zhou, An invocation cost optimization method for web services in cloud environment. Scientific Programming **2017**(4358536), 9 (2017)
35. Y. Chen, J. Huang, C. Lin, J. Hu, A partial selection methodology for efficient QoS-aware service composition. IEEE Transactions on Services Computing **8**(3), 384–397 (2015)
36. L. Qi, W. Dou, J. Chen, Weighted principal component analysis-based service selection method for multimedia services in cloud. Computing **98**(1), 195–214 (2016)
37. X. Xu, Q. Cai, G. Zhang, J. Zhang, W. Tian, X. Zhang, A.X. Liu, An incentive mechanism for crowdsourcing markets with social welfare maximization in cloud-edge computing. Concurrency and Computation: Practice and Experience (2018). https://doi.org/10.1002/cpe.4961

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.