Incorporating Reverse Search for Friend Recommendation with Random Walk

Qing Yang[1], Haiyang Wang[1], Mengyang Bian[1], Yuming Lin[2], and Jingwei Zhang[2]

[1]Guangxi Key Laboratory of Automatic Measurement Technology and Instrument, Guilin University of Electronic Technology, China

[2]Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, China

Abstract: *Recommending friends is an important mechanism for social networks to enhance their vitality and attractions to users. The huge user base as well as the sparse user relationships give great challenges to propose friends on social networks. Random walk is a classic strategy for recommendations, which provides a feasible solution for the above challenges. However, most of the existing recommendation methods based on random walk are only weighing the forward search, which ignore the significance of reverse social relationships. In this paper, we proposed a method to recommend friends by integrating reverse search into random walk. First, we introduced the FP-Growth algorithm to construct both web graphs of social networks and their corresponding transition probability matrix. Second, we defined the reverse search strategy to include the reverse social influences and to collaborate with random walk for recommending friends. The proposed model both optimized the transition probability matrix and improved the search mode to provide better recommendation performance. Experimental results on real datasets showed that the proposed method performs better than the naive random walk method which considered the forward search mode only.*

Keywords: *Social networks, friend recommendation, reverse search.*

1. Introduction

Social networks are very popular in a variety of fields, which have been blended in with people's lives and are playing important roles. At the same time, those large-scale social networks are also facing a big challenge on improving their social activity and user loyalty. Friend recommendation is an important mechanism to make social networks more popular and to provide a better user experience, including strengthening user vitality. For example, recommending backpackers for self-help tourists can enhance travel information sharing and travel experience, recommending scholars on academic social networks is conducive to broaden their research insights and to promote academic exchanges.

Due to the constraints of huge social network scale coupled with the sparsity of relationships between users, efficient and accurate friend recommendation on social networks presents significant challenges [23]. Aiming at the above focuses, some studies [14, 22, 25, 28] found that it can produce effective recommendation results by transforming social networks into web graphs and then applying random walk strategy. The typical recommendation model by random walk on graphs is to set up web graphs on social networks firstly, which are composed of nodes and edges. In web graphs, a node represents a social user and an edge represents a social connections between two specific users. Then the recommendation process will work by computing the arrival probability from a target node to the remaining nodes on graphs. The probability distribution after a few walks reflects the influence that are exerted by the target node on the other nodes. A node with a high probability value implies that the corresponding user is more likely to become a friend with the target user. As a result, the top-k users under the descent ranking on arrival probability can be recommended to the target user. Here, the primary assumption for friend recommendation is, if there is a "convenient" access path from the target user to some user, they are easy to become friends. We call this access convenience as their social influence.

In above models for friend recommendation, the social influence is exerted by the target node to have an effect on the other nodes, which is known as forward social influence. But in our real life, the target user is also prone to establish contact with those people with popular influence in some fields, and even the target user does not have a convenient access path to them or may not know them in person at the moment. A recommendation application should not only consider the influence from the target user to other users, but also should consider how other users can affect the target user. The latter is called as reverse social influence. For example, when you plan to travel in some place, it will be more reasonable to recommend somebody who know that place well but are not very familiar with you than those whom you

have regular contacts with but know that place little. Obviously, it is possible to create many novelty ideas if a new scholar steps into an academic circle with overlapping research interests by recommendation.

This paper focuses on providing more effective methods for friend recommendations, and a comprehensive consideration on both forward social influence and reverse social influence are exploited to propose accurate friendships. The main contributions are,

- Introducing FP-Growth method for constructing transition probability matrix and providing a statistical representation for user interactions, which can create a more effective data basis for the following recommendation computation.
- Defining reverse social influence and integrating reverse search model with random walk to improve friend recommendation performance.
- Designing and conducting experiments on the real data sets to verify the effectiveness of our proposed recommendation model.

The main body of this paper is organized as follows. A summary of the related study on friend recommendation is presented in section 2. Section 3 and 4 elaborate the proposed recommendation method. The experimental datasets as well as experimental analysis are discussed in section 5. At last, we conclude the study and discuss the future work.

2. Related Work

Based on both the properties of recommended items and application purposes, there are two kinds of recommendation tasks, namely recommending users and recommending users' attachments. The former is a coarse-grained application and is often applied on social networks, and friend recommendation is just this case. The latter often works on users' interests, such as films, goods, tourism and other specific items [16].

An early strategy for recommending friends is to model user interaction frequency into different closeness levels between users. Lo and Lin [15] proposed a friend recommendation method based on two basic assumptions, one is that the interaction frequency is proportional to the closeness between users, the other is those users with more common friends are apt to establish new friendships. Chin [4] studied the impact of the interaction time on mobile communication devices for building friendships. One obvious deficiency of the above methods is that they failed to balance the impact of negative interaction between users. Shen *et al.* [20] focused the blog network for friend discovery. They established users' interest model through mining text information from blogs, and then discovered potential friends based on user similarity. Zheng *et al.* [30] established GeoLife 2.0 for personalized friend recommendation, which

utilized GPS-data-driven social networking service and users' historical trajectory to measure user similarity. [26, 27] studied the collaborative model between different nodes in a network to serve for communications. Faisal *et al.* [5] exploited the user reputation to discover experts in online discussion forums.

The above recommendation methods often rely on the local features, which cannot perform well when facing a large user base and sparse relationships. PageRank [2] and its variants [3, 6], who were initially designed for ranking web pages, are now applied to improve recommendation performance. The core of personalized PageRank is just a Random Walk process with Restart (RWR) [19]. For every step, RWR allows walkers to return to the original position with an empirical ratio, which can hold the overall information and deal with sparsity effectively [1]. Some studies have contributed effective algorithms for friend recommendation by improving random walk with restart on social networks [17, 18, 24, 29]. Yan *et al.* [28] focused on the cold-start problem of friend recommendation by introducing cross-platform social relationships and user behaviours, which made a performance gain for their method based on random walk. Xia *et al.* [25] presented an innovative method by improving RWR to recommend co-authors for scholars, which took three academic factors into consideration, namely co-author order, latest collaboration time and times of collaboration, to achieve a comprehensive improvement on the precision, recall and coverage of recommendation. Inspired by traditional link prediction, Liu *et al.* [14] came up with a personalized link prediction method with local random walk on social networks, which ensured each individual to get an equal number of diversified predictions. Wang *et al.* [22] integrated uniform vertex sampling into random walk technique to characterize properties of user pairs and to achieve both accuracy and efficiency on recommendation.

The above recommendation methods evolved from RWR are based on the forward search strategy, which pay little attention on the exploration of the reverse social influence between users. This paper will focus on exploiting and integrating the reverse social influence into RWR for more accurate friend recommendation ranking, which will work on a transition probability matrix initialized by FP-Growth algorithm [7].

3. Constructing Transition Probability Matrix on FP-Growth

The proposed recommendation method is detailed in the following two sections. We reconstruct the transition probability matrix based on social relationships and FP-Growth algorithm, and then define reverse social influence to integrate reverse

search with random walk for recommendations. The two stages complement each other, the former guarantees that the users having constant direct or indirect contacts with the target user will be recommended with high probability, and the latter ensures the users with great influence gain the chance to be recommended.

3.1. Representing Social Networks

In brief, a social network is deemed as a web graph consisting of nodes and edges, which can be represented by $G=(V,E)$, where V is the set of n vertices, each node represents a user, and E is the set of y edges, each corresponds to the social relationship between a pair of users. The set of edges in G can be represented by the adjacency matrix A as Equation (1).

$$a_{i,j} = \begin{cases} 1 & < i, j > \in E \\ 0 & otherwise \end{cases} \quad (1)$$

Where i and j $(i, j \in V, i \neq j)$ are two arbitrary nodes of G. If $a_{i,j}=1$, an edge is associated with i and j, namely a social relationship exists between the corresponding user pair. If $a_{i,j}=0$, there is no edge between i and j, namely no social relationship exists between the corresponding users. For any node i in G, the out-degree of i is equal to the number of edges starting from i, denoted as $d_{od}(i) = \sum_{j=1}^{n} a_{i,j}$. The in-degree of i is indicated by the number of edges ending at i, denoted as $d_{id}(i) = \sum_{j=1}^{n} a_{j,i}$. If each edge is accompanied by a value $p_{i,j}$, which represents the probability of the direct social influence exerted on j by i, a transfer probability matrix P can be constructed. In the initial status, $p_{i,j}$ is computed as Equation (2).

$$p_{i,j} = \frac{a_{i,j}}{d_{od}(i)}, \forall i, j \in V, i \neq j \quad (2)$$

Based on the initial matrix, friend recommendation can be defined to discover a user ranking by computing closeness between a target user and all other users. Obviously, the initial status of P will have a great influence on the closeness computing. Here, closeness means possibilities that a user should be recommended to the target user.

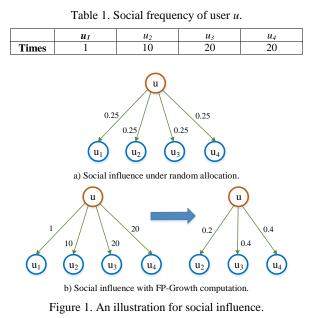## 3.2. Reconstructing Transition Probability Matrix by FP-Growth

Equation (2) indicates that the initial transition probability of the target node is decided by its out-degree, which does not give a full consideration of the real situations between users. For example, given a target user u, we suppose it interacts with users u_1, u_2, u_3, u_4. According to the Equation (2), we will construct the web graph as shown in Figure 1-a, in which the transition probabilities of u to $u_1 \sim u_4$ are equal and the closeness of $u_1 \sim u_4$ are also equivalent for u. This is evidently inappropriate since the social frequency of u to $u_1 \sim u_4$ may not be necessarily equal. Naturally, the users with different social frequency should have different closeness to u.

To compute users' closeness to a specific user efficiently, we introduce FP-Growth algorithm to construct the web graph corresponding to a social network. FP-Growth is initially adopted for mining frequent item sets, here we utilize it to construct a FP-tree structure to discover a set of users who frequently interact. The advantage of FP-Growth is to scan the dataset only two times, and to compress the data into a FP-tree structure to make efficiency gains when users with high influence are retained.

There are two stages to reconstruct the transition probability matrix with FP-Growth. The first stage is to construct a FP-tree and to find frequent user sets when scanning the database. Each user is associated with a value, $sup(i)$, which corresponds to the user frequency and decides the content of FP-tree under a preset minimum threshold. Then a conditional pattern tree is created for each user by combining paths in the FP-tree. Finally, all the frequent patterns of each user will be generated through a recursive traversal process on all conditional pattern trees. The second stage is to reconstruct the transition probability matrix. The outputted frequent item sets for each user in the first stage can be computed as the interaction frequency between different user pairs, denoted as $sup(i, k)$. We only need to assign the values between user pairs, $sup(i, k)$, to the corresponding edges in the web graph, namely $a_{i,k}=w_{i,k}=sup(i,k)$. Thus the reconstructed transition probability matrix P for the web graph can be represented in Equation (3).

$$p_{i,k} = \frac{w_{i,k}}{\sum_{j=1}^{n} w_{i,j}}, \forall i, k \in V, i \neq k \quad (3)$$

Assuming the total interaction frequency of u with the above four users is 51, the minimum threshold is 4. Table 1 lists the social details between u and other users. It is obvious that the influence from u to u_1 can be ignored since it cannot satisfy the threshold requirement. This will also be beneficial for the following recommendation computation since it is not a hot line for u. In fact, this kind of infrequent social interaction in the complex network environment is likely to be some advertising promotion, or even some other occasional information, which will cause performance degradation for building friendships between users. Based on the above method, we can reconstruct a web graph as Figure 1-b. Compared with Figure 1-a, the current model can present a more reasonable representation for social influence implied in the data.

Table 1. Social frequency of user *u*.

	u_1	u_2	u_3	u_4
Times	1	10	20	20

a) Social influence under random allocation.

b) Social influence with FP-Growth computation.

Figure 1. An illustration for social influence.

4. Integrating Reverse Social Influence into Random Walk

Random walk is a popular strategy to discover potential friends for a specific user in graph structure, which holds all possibilities of each walking step from the current status to the next status in a transition probability matrix *P*. The whole process can be formalized as Equation (4), where *i* is an initial node, P^T holds the transpose of *P*, $x_i^{(t)}$ indicates a column vector corresponding to the probability distribution that the walker arrives at all other nodes on the web graph in step *t*. More specially, the influence that *i* exerts on *j* at step *t* can be expressed as $x_i(j,t)$. When *t* is large enough, $x_i^{(t)}$ will tend to be stabilized. The final vector will decide the user ranking on closeness for user *i*.

$$X_i^{(t+1)} = P^T X_i^{(t)} \qquad (4)$$

$$X_i^{(t+1)} = \alpha P^T X_i^{(t)} + (1 - \alpha)X_i^{(0)} \qquad (5)$$

Considering the spiders on graphs, Equation (5) presents a variant of random walk, which gives the walking process a restart opportunity in each step, namely RWR. As shown in Equation (5), the restart ratio is decided by an empirical parameter α, and $x_i(0)$ is the initial vector constructed in section 3. RWR can bring the walker out in certain steps on the preset parameter α when the walker is stuck in the local circulation or absorbing nodes, namely spiders. In addition, Li *et al.* [12] proposed a Springy Random Walk With Restart model (SRWR), which defined the restart parameter as $e^{-\gamma t}$, where γ indicates a springy ratio and *t* stands for random walk length. Obviously, the restart ratio $(1 - e^{-\gamma t})$ is positively correlated with random walk length *t*, i.e., when *t* becomes larger, the walker will gain more proportion to return through the dynamic adjustment mechanism of the ratio. The friendship building will benefit substantially from the above mechanism. On a web graph transformed from a social network, since the walkers' behaviours are decided by the social influence between users, the vector x_i just represents the social influence distribution from the target node *i* to all other nodes when a steady state is reached. This will provide the basis for friend recommendation.

However, in many practical situations, the social influence exerted on a user by the target user should be enhanced when considering recommendations, for example, the reverse social influence should also be covered. Here, we will detail the computation of the reverse social influence distribution based on reverse search and its integration with RWR for friend recommendations. For a given web graph *G* and a specific target node *i*, the reverse social influence of *i* means the influence exerted by all other nodes to *i*, denoted as $x_i^{Reverse}$, which is defined in Equation (6), where *t* is the current random walk length. Any reverse social influence on oneself is zero since self-recommendation is meaningless. Since the reverse search process should be executed for every other node to match the target node, namely *n-1* times, the time complexity of the reverse search is *n* times of the forward search.

$$X_i^{Reverse}(j, t) = x_j(i, t), \forall j \in V, j \neq i \qquad (6)$$

In order to balance the forward and reverse influence between users for friend recommendation, we introduce a balance factor β and define *Reci(j)* to indicate the closeness score that *j* should be recommended to the target user *i*, details are presented in Equation (7) where the random walk length *t* is assumed to be certain. One can achieve optimal results for different application scenarios through adjusting the parameter β. All nodes will be ranked in a descending order according to the closeness score, and then those users corresponding to the top-*k* nodes will be outputted.

$$\begin{aligned} \mathrm{Re}\, c_i(j) &= \beta x_i(j) + (1 - \beta)x_i^{Reverse}(j) \\ \forall j &\in V, j \neq i \end{aligned} \qquad (7)$$

Algorithm 1 describes the details of the proposed friend recommendation method based on the integration of reverse search with random walk.

Algorithm (1): friend recommendation by integrating reverse search into random walk.

Input: a social network dataset, D
*   a target node, i*
*   frequency threshold: minsup*
*   restart ratio: α*
*   balance factor: β*
*   number of recommended friends: k*
Output: a user set with the highest k closeness scores, E_n;
Steps:

1. *scanning D and finding out frequent user sets by minsup to establish FP-tree;*
2. *for each user j in the FP-tree, computing sup(i, j);*
3. *attaching sup(i, j) to the corresponding edges of G(V,E) and constructing the transition probability matrix P according to Formula. 3;*
4. *while* $x_i^{(t+1)} - x_i^{(t)} > \varepsilon$ *do*
5. $x_i^{(t+1)} = \alpha P^T x_i^{(t)} + (1-\alpha)x_i^{(0)}$
6. *end while*
7. *for each* $j \in V$, $j \neq i$ *do*
8. $\text{Re}\, c_i(j) = \beta x_i(j) + (1-\beta)x_i^{\text{Re}verse}(j)$
9. *end for*
10. *ranking all nodes in descending order on Rec$_i$(j) and assigning the top k nodes to E$_n$;*

5. Experiments

5.1. Datasets

We conduct experiments to test the proposed recommendation method on three datasets: Digital Bibliography and Library Project (DBLP), Slashdot and Epinions. DBLP [11] is a computer literature dataset, from which we establish an academic social network according to collaboration among authors. The latest DBLP dataset contains 1.4 million authors along with 3.4 million publications. Slashdot is a website focusing on information technology and can be viewed as a kind of blog, where users can make comments on news and blogs. The dataset of Slashdot is composed of a total of 82168 users and 948464 links [9]. Epinions is a website for popular consumer reviews, on which users can choose other users to trust or to be trusted. The dataset of Epinions consists of 75879 users and 508837 links [21]. The above three datasets can help us construct social networks in different fields to test the performance of the proposed method. After applying the FP-Growth process, the details of the constructed web graphs are presented in Table 2, in which the frequency threshold minsup is set to 4 based on the comprehensive compromise of both the scale of social networks and the statistics of user interactions.

Table 2. Details of datasets through FP-Growth.

	Number of nodes	Number of edges	Average degree
DBLP	13345	24738	1.85
Slashdot	26335	100058	3.80
Epinions	10879	79988	7.35

5.2. Evaluation Metrics

We use two metrics for quantifying the performance of the proposed method on friend recommendations, Accuracy (Acc) [13] and the Mean Reciprocal Rank (MRR) [8]. The Acc metric simply evaluates the algorithm's performance by the quantity of correct recommendations, defined as Equation (8), where *k* is the number of all recommended friends, and *f* is the number of correct recommendations in the final list.

Apparently a higher Acc value means a better performance. The MRR metric evaluates the algorithm's performance by computing the ranking accuracy of friends, as shown in the Equation 9, where L_i is the ranking of a friend that should be recommended in the final list. Since a ranking near the top will contribute a larger value to MRR, a higher MRR indicates a better recommendation performance. In addition, in order to establish a set of potential friends that should be recommended preferentially in future for trustworthy verification, we execute a filtering process to select the users, which have the most frequent contact with the target user in the last period of time, as the seed set.

$$Acc = \frac{f}{k} \qquad (8)$$

$$MRR = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{L_i} \qquad (9)$$

5.3. Experimental Results

Konstas *et al*. [10] provide a good proposal for testing the optimal value of the restart ratio α, which should be between 0.7 to 0.9. In our experiments, a high α will be helpful to reach the steady state with less computation because of fewer spiders, which is improved by the FP-Growth process. We set α as 0.9 and the balance ratio β as 0.5. For each dataset, we randomly take 20 target users for verification, and then calculate the mean value of the experimental outputs on both Acc and MRR. The length of recommended list *k* is limited to 15.

To present the performance of our proposed method, we make comparisons with personalized PageRank [19] and pure reverse search method (see Equation (6)). Figure 2 shows the experimental results on Acc, which indicates that our method (abbr. as PageRank Reverse Search (PRRS) performs better than Personalized PageRank (abbr. as PR) and pure reverse search (abbr. as RS). Though reverse search does not provide more information for recommendation than personalized PageRank, their integration still makes a significant improvement for recommendation on different kinds of social applications. Reverse social influence creates new opportunities for building friendships that cannot be easily reached by the forward search.
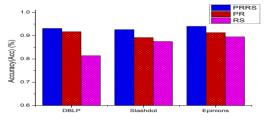

Figure 2. Performance comparison on Acc.

Table 3 presents the experimental results on three datasets under the MRR metric, which shows that our method still prevails over personalized PageRank and pure reverse search method in most of the tests. Reverse search has some advantage indeed on DBLP dataset, the reason tends to be that reverse influence has a substantial impact on friendship building in academic circles, this is also why we consider to incorporate reverse social influence into our model and eventually gain better performance on other general cases. To be specific, the proposed recommendation method benefits from the integration of random walk and reverse search, and is effective for different social applications. A potential friend that should be recommended gains a larger possibility to be included in the final recommended list and appear at the top of the list under the proposed method.

Table 3. Performance comparison on MRR.

	PRRS	PR	RS
DBLP	0.449	0.444	0.471
Slashdot	0.315	0.304	0.288
Epinions	0.243	0.240	0.236

6. Conclusions

This study contributed an improved method for recommending friends, which covered reverse social influence to be integrated into random walk and to make full use of the advantages of both forward search and reverse search. FP-Growth algorithm was firstly executed to collect the frequency information of user interactions for constructing web graph model of social networks as well as reconstructing the transition probability matrix. Then the reverse search model was defined and integrated with random walk for recommendations. We designed experiments on different datasets to verify the proposed method and to make comparisons with personalized PageRank and pure reverse search model. Experimental results on two metrics, accuracy as well as mean reciprocal rank, showed that the integrated model was more effective.

Two primary factors made the proposed method to be more effective on recommending friends. One was to reconstruct the transition probability matrix based on FP-Growth to present intuitive social information, the other was to take the reverse social influence into account. The above two factors were both computed from a global perspective. In future, we plan to combine local Bayesian inference strategy with random walk for further improvement on both recommendation computation performance and local recommendation accuracy.

Acknowledgements

References

[1] Bagci H. and Karagoz P., "Context-Aware Friend Recommendation for Location Based Social Networks Using Random Walk," *in Proceedings of the 25th International Conference Companion on World Wide Web*, Montreal, pp. 531-536, 2016.

[2] Brin S. and Page L., "Anatomy of a Large-Scale Hypertextual Web Search Engine," *in Proceedings of the 7th International World Wide Web Conference*, Brisbane, pp.107-117, 1998.

[3] Broder A., Kumar R., Maghoul., Raghavan P., Rajagopalan S., Stata R., Tomkins A., and Wiener J., "Graph Structure in the Web," *Computer Networks*, vol. 33, no. 1-6, pp. 309-320, 2000.

[4] Chin A., "Finding Cohesive Subgroups and Relevant Members in the Nokia Friend View Mobile Social Network," *in Proceedings of International Conference on Computational Science and Engineering*, Vancouver, pp. 278-283, 2009.

[5] Faisal M., Daud A., and Akram A., "Expert Ranking using Reputation and Answer Quality of Co-Existing Users," *The International Arab Journal of Information Technology*, vol. 14, no. 2, pp. 118-126, 2017.

[6] Gyongi Z., Garcia-Molina H., and Pedersen J., "Combating Link Spam with Trustrank," *in Proceedings of 30th International Conference on Very Large Databases*, Toronto, pp. 576-587, 2004.

[7] Han J., Pei J., Yin Y., and Mao R., "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53-87, 2004.

[8] Huang W., Kataria S., Caragea C., Mitra P., and Giles L., "Recommending Citations: Translating Papers into References," *in Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, Hawaii, pp. 1910-1914, 2012.

[9] Kandhway K. and Kuri J., "Using Node Centrality and Optimal Control to Maximize Information Diffusion in Social Networks," *IEEE Transactions on Systems, Man, and Cybernetics System*, vol. 47, no. 7, pp. 1099-1110, 2017.

[10] Konstas I., Stathopoulos V., and Jose J., "On Social Networks and Collaborative Recommendation," *in Proceedings of the 32nd International ACM SIGIR Conference on*

Research and Development in Information Retrieval, Boston, pp. 195-202, 2009.

[11] Le T. and Zhang D., "DBLPminer: A Tool for Exploring Bibliographic Data," *in Proceedings of the 16th IEEE International Conference on Information Reuse and Integration*, San Francisco, pp. 435-442, 2015.

[12] Li H., He K., Wang J., and Peng Z., "A Friends Recommendation Algorithm Based on Formal Concept Analysis and Random Walk in Social Network," *Journal of Sichuan University (Engineering Science Edition)*, vol. 47, no. 6, pp. 131-138, 2015.

[13] Likavec S., Osborne F., and Cena F., "Property-based Semantic Similarity and Relatedness for Improving Recommendation Accuracy and Diversity," *International Journal on Semantic Web and Information Systems*, vol. 11, no. 4, pp. 1-40, 2015.

[14] Liu J., Zhu Y., and Zhou T., "Improving Personalized Link Prediction by Hybrid Diffusion," *Physica A: Statistical Mechanics and its Applications*, vol. 447, pp. 199-207, 2016.

[15] Lo S. and Lin C., "WMR-A Graph-Based Algorithm for Friend Recommendation," *in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, Hongkong, pp. 121-128, 2006.

[16] Massa P. and Bhattacharjee B., "Using Trust in Recommender System: An Experimental Analysis," *in Proceedings of the 2nd International Conference on Trust Management*, Oxford, pp. 221-235, 2004.

[17] Mourchid F. and Koutbi M., "LBRW: A Learning based Random Walk for Recommender Systems," *International Journal of Information Systems and Social Change*, vol. 6, no. 3, pp. 15-34, 2015.

[18] Nie D., Fu Y., Zhou J., Liu Z., Zhang Z., and Liu C., "A Personalized Recommendation Algorithm Via Biased Random Walk," *in Proceedings of the 11th International Joint Conference on Computer Science and Software Engineering*, Chon Buri, pp. 292-296, 2014.

[19] Page L., Brin S., Motwani R., and Winograd T., "The PageRank Citation Ranking: Bringing order to the Web," Technical Report, Stanford InfoLab, 1999.

[20] Shen D., Sun J., Yang Q., and Chen Z., "Latent Friend Mining from Blog Data," *in Proceedings of the 6th International Conference on Data Mining*, Hongkong, pp. 552-561, 2006.

[21] Simpson M., Srinivasan V., and Thomo A., "Clearing Contamination in Large Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1435-1448, 2016.

[22] Wang P., Zhao J., Lui J., Towsley D., and Guan X., "Unbiased Characterization of Node Pairs over Large Graphs," *ACM Transactions on*

Knowledge Discovery from Data, vol. 9, no. 3, pp. 22, 2015.

[23] Wang Y., Li J., Liu Q., and Ren Y., "Prediction of Purchase Behaviors Across Heterogeneous Social Networks," *The Journal of Supercomputing*, vol. 71, no. 9, pp. 3320-3336, 2015.

[24] Wang Z., Tan Y., and Zhang M., "Graph-based Recommendation on Social Networks," *in Proceedings of the 12th International Asia-Pacific Web Conference*, Busan, pp. 116-122, 2010.

[25] Xia F., Chen Z., Wang W., and Li J., "MVCWalker: Random Walk-Based Most Valuable Collaborators Recommendation Exploiting Academic Factors," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 364-375, 2015.

[26] Xu L. and Gulliver T., "Performance Analysis for M2M Video Transmission Cooperative Networks Using Transmit Antenna Selection," *Multimedia Tools and Applications*, vol. 76, no. 22, pp. 23891-23902, 2017.

[27] Xu L., Wang J., Zhang H., and Gulliver T., "Performance Analysis of IAF Relaying Mobile D2D Cooperative Networks," *Journal of the Franklin Institute*, vol. 354, no. 2, pp. 902-916, 2017.

[28] Yan M., Sang J., Mei T., and Xu C., "Friend Transfer: Cold-start Friend Recommendation with Cross-Platform Transfer Learning of Social Knowledge," *in Proceedings of the International Conference on Multimedia and Expo*, San Jose, pp. 1-6, 2013.

[29] Ying J.J.C., Kuo W., Tseng V., and Lu E., "Mining User Check-In Behavior with a Random Walk for Urban Point-of-Interest Recommendations," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, pp. 40, 2014.

[30] Zheng Y., Chen Y., Xie X., and Ma W., "GeoLife2.0: A Location-Based Social Networking Service," *in Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, Taipei, pp. 357-358, 2009.

Qing Yang is an associate professor at Guilin University of Electronic Technology, China. Her research interests include intelligent information processing, social network analysis, and large-scale data processing optimization.

Haiyang Wang is a M.S., born in 1981. His research interests includes personalized recommendation and applications.

Mengyang Bian is a M.S, born in 1992. His research interests include data mining and recommendation optimization.

Yuming Lin is an associate professor at Guilin University of Electronic Technology, China. He obtained his Ph.D from East China Normal University, China in 2013. His research interests include sentiment analysis, Web data mining and knowledge graph.

Jingwei Zhang is an associate professor at Guilin University of Electronic Technology, China. He obtained his Ph.D from East China Normal University, China in 2012. His research interests include massive data management, distributed computing framework and big data analytics for emerging applications.