

---

# Exploiting Direct And Indirect Information For Friend Suggestion In ZingMe

---

**Kien Duy Nguyen**

School of Information and Communication Technology  
Hanoi University Of Science and Technology  
Hanoi, Vietnam  
duykienvp@gmail.com

**Quang Nhat Nguyen**

School of Information and Communication Technology  
Hanoi University Of Science and Technology  
Hanoi, Vietnam  
quang.nguyennhat@hust.edu.vn

**Tuan Pham Minh**

VNG Corporation  
Hanoi, Vietnam  
tuanpm@vng.com.vn

**Thanh Trung Nguyen**

VNG Corporation  
Hanoi, Vietnam  
thanhnt@vng.com.vn

## Abstract

Friend suggestion is a fundamental problem in social networks with the goal of assisting users in creating more relationships, and thereby enhances interest of users to the social networks. This problem is often considered to be the link prediction problem in the network. ZingMe is one of the largest social networks in Vietnam. In this paper, we analyze the current approach for the friend suggestion problem in ZingMe, showing its limitations and disadvantages. We propose a new efficient approach for friend suggestion that uses information from the network structure, attributes and interactions of users to create resources for the evaluation of friend connection amongst users. Friend connection is evaluated exploiting both direct communication between the users and information from other ones in the network. The proposed approach has been implemented in a new system version of ZingMe. We conducted experiments, exploiting a dataset derived from the users' real use of ZingMe, to compare the newly proposed approach to the current approach and some well-known ones for the accuracy of friend suggestion. The experimental results show that the newly proposed approach outperforms the current one, i.e., by an increase of 7% to 98% on average in the friend suggestion accuracy. The proposed approach also outperforms other ones for users who have a small number of friends with improvements from 20% to 85% on average. In this paper, we also discuss a number of open issues and possible improvements for the proposed approach.

## 1 Introduction

Online social networks bring people a new way to receive, to exchange and to share information. In order to attract and keep interest of users, one important problem in social networks is how to make virtual society of users bigger and their relationships closer. Relationships can be friendships, profession links, collaborations and so on. Suggesting new relationships for users is one way of tackling this. We can define this problem as follows: given a snapshot of the social network at time  $t$  and for each user, we want to predict new relationships that will appear in the future [1]. This problem is often considered as link prediction or link recommendation problem: given a snapshot of a network at time  $t$ , we want to output a predicted list of edges that will be added to the network in the future.

There is one question we should consider for friend suggestion problem: can new friendships of users be predicted accurately based on information in the network [12]? Many reasons may cause new friendships between users. For instance, they can meet each other in a party and then make friends in the network. However, we can guess that when they come to a party, they properly have some mutual friends or work for same company. If we can obtain that information, we can have foundation to believe in success of predictions.

Another challenge is a massive class skew [2]. Data from ZingMe shows that only 0.02% of possible friendships are actually established during the period of data collection. This causes a difficulty in distinguishing established friendships from non-established ones.

ZingMe is one of the largest social networks in Vietnam with about 8 millions active users per month. In ZingMe, we call this problem friend suggestion problem because relationships among users are friendships. ZingMe currently has an approach to solve this problem but it has some limitations and disadvantages. Although current approach uses some important information sources to access ability to create friendship between two users, there are still other ones ignored. In addition, it uses only information between two users, wastefully discarding information from other users in the network which can also affect to progress of creating friendship between them. This leads to the need for a new approach that can efficiently exploit information sources of this social network.

Our work presents a novel approach for friend suggestion problem on social networks that can effectively exploit information between two users and information from other users in the network in suggestion process. Our approach also leverages many different knowledge sources such as network structure, attributes of vertices and attributes of edges. Our work also brings users of ZingMe a new friend recommender system that captures better their interests in the network. We also anonymize and give our dataset available to research community.

The rest of this paper is organized as follows. Section 2 presents current approach deployed for ZingMe. Our proposed approach is presented in Section 3. Some experiments and discussions are given in Section 4. Related works are summarized in Section 5. Finally, Section 6 concludes this paper and shows some future directions.

## 2 The Friend Suggestion Approach Currently Used in ZingMe

In ZingMe, each user may have a list of friends, a list of IP addresses he used to log in to ZingMe, a list of schools and a list of companies he have studied or worked. Currently, a approach deployed to solve friend suggestion problem in ZingMe that uses following features between two users: *number of mutual friends*, *number of mutual schools*, *number of mutual companies* and *number of mutual IP addresses*.

For a target user  $u$  and a candidate  $v$ , the current approach builds a graph to assess ability to form friendship from  $u$  to  $v$ . Given  $N_u$  as number of friends of  $u$ , the graph includes four vertices and edges presenting strength of relationships amongst those vertices. The vertices have  $t_1, t_2, t_3, t_4$  as weights when  $u$  and  $v$  have or do not have mutual friends, mutual schools, mutual companies, mutual IP addresses, respectively.  $t_i = 1$  or  $t_i = w_i > 1$  when  $u$  and  $v$  have or do not have mutual information for feature  $i$ . A vertex  $i$  is "turned on" if  $u$  and  $v$  have mutual information for that vertex. Especially, vertex 1 is always turned on.

Edge  $(i, j)$  has weight  $e_{i,j}$  that indicates weight when both of  $t_i$  and  $t_j$  are turned on, and  $e_{i,j}$  is turned on, otherwise, is equal to 1.0. For example, if we think that for two users, having mutual companies and mutual IP addresses is more important than having mutual schools and IP addresses, we may set  $e_{3,4} > e_{2,4}$ . Given  $n$  as number of turned on edges, function for accessing ability to make friends between  $u$  and  $v$  as follows:

$$score_{u,v} = n * \sum_{i,j=1}^4 e_{i,j} * t_i * t_j \quad (1)$$

We notice that the current approach only uses some knowledge sources in the network for friend suggestion and neglects many potential sources such as groups of users or interactions of users. It also creates strong ties among features through edges of the graph, thereby creating difficulties in

adding more features. In addition, it only exploits direct information between two users, discarding information from other ones in the network.

### 3 The Proposed Friend Suggestion Approach

#### 3.1 Formal Definitions

Social network is modelled as directed graph  $G = (V, E)$  in which each vertex  $u \in V$  represents a user in the network and each edge  $e_{u,v} \in E$  represents relationship:  $v$  is a friend of  $u$ . Because of symmetry in friendship in social network ZingMe, if  $e_{u,v}$  exists,  $e_{v,u}$  also exists. Weight  $c_{u,v}$  represents strength of friendship between  $u$  and  $v$ .  $t_{u,v}$  or  $t_e$  is the time when edge  $e_{u,v}$  is established.

For times  $t, t_1, t_2$ , let  $G[t] \subseteq G$  and  $G[t_1, t_2] \subseteq G$  consist of edges with  $t_e \leq t$  and  $t_1 \leq t_e \leq t_2$ , respectively. The link prediction problem for social networks can be defined as follow: given times  $t_0 < t'_0 \leq t_1 < t'_1$ , based on  $G[t_0, t'_0]$ , we want to output a list of edges which are not in  $G[t_0, t'_0]$  and are predicted to present in  $G[t_1, t'_1]$  [1].

#### 3.2 Knowledge Sources

The proposed friend suggestion approach uses knowledge sources: *network structure*, *attributes of vertices*, for example, age, sex or education and *attributes of edges*, for example, interactions when a user comments on a picture, likes a page or sends a message

Specific features are derived from these knowledge sources with assumption that if two vertices are not connected, the more mutual information they have, the more probability to be connected they have, and if they are already connected, it represents strength of their connection which is greater when they have more mutual information. Simultaneously, type and reliability of data are also considered when choosing these features.

For the graph structure, *the number of mutual adjacent vertices* is the feature derived and equal to  $|N_u \cap N_v|$ . Data from ZingMe and Facebook [12] also show that more than a half of new friendships are established from users having mutual friends.

For the attributes of vertices, selected features are *number of mutual schools*, *number of mutual groups* and *number of mutual IP addresses*.

For the attributes of edges, the feature chosen is *number of "mutual interactions"*. For instance, if two users like same picture, that can be considered as a mutual interaction between them. There are many types of mutual interactions such as users are tagged in same pictures, comment on same posts, and it may be better to consider each mutual interaction as an individual feature but they are considered as one because of computing capability of real system.

Our approach reuses some features from current approach, including number of mutual adjacent vertices, number of mutual schools and number of mutual IP addresses. It also adds features derived from groups and interactions of users, which were neglected in the current approach. It eliminates the feature relating to companies of users because of unreliability of this feature. In ZingMe, users provide names of their companies by free text without any standardization. By manually examining content of those names, we found them unreliable.

#### 3.3 The Proposed Approach

The proposed approach for predicting new connections for a vertex  $u$  is based on accessing ability to form connections between  $u$  and other ones. Given a candidate  $v$ , it used direct information between  $u$  and  $v$ , along with indirect information from other vertices. These types of information are obtained from *affinity* between two vertices.

As illustrated in Figure 1, proposed approach consists of two components: *affinity calculation* and *suggestion calculation*. Suggestion calculation component includes three phases to calculate suggestions for a target vertex  $u$ : candidate selection phase finds potential candidates to form edges to  $u$ ; score indicating ability to form edge to  $u$  of each candidate is calculated in score calculation phase; ranking phase ranks candidates to output a suggestion list for  $u$ .

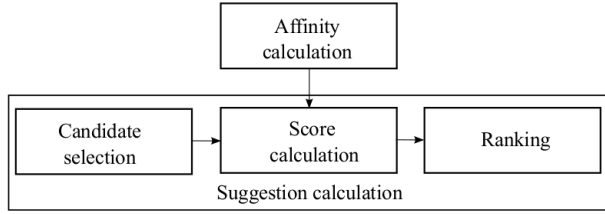


Figure 1: Proposed approach diagram

### 3.3.1 Affinity Calculation

Affinity of  $u$  to  $v$ ,  $aff_{u,v}$  indicates how  $u$  is interested in or affined to  $v$  and is used as  $c_{u,v}$  on the graph. It is noted that  $aff_{u,v}$  may not be equal to  $aff_{v,u}$ . In order to calculate  $aff_{u,v}$ , features are used based on the hypothesis that the more mutual information  $u$  and  $v$  have, the higher  $aff_{u,v}$  is. In addition, each feature has its own level of influence to  $aff_{u,v}$ , i.e., weight. The computation of  $aff_{u,v}$  is defined as follows:

$$aff_{u,v} = \sum_{i=1}^{Nf} w_i * \log(S_i + 1) \quad (2)$$

in which  $Nf$  is the number of features,  $w_i$  is the weight for feature  $i$  and  $S_i$  is the value of feature  $i$ . The logarithm function is used for normalization because for different features, their values  $S_i$  may belong to very different ranges. For example, the number of mutual schools between two users is often smaller than 3 while the number of mutual interactions may get value at tens or hundreds. The logarithm function helps features get in closer ranges.

### 3.3.2 Suggestion Calculation

**Candidate Selection** The goal of the candidate selection phase is to select potential candidates to connect to target vertex  $u$ . This issue arises when number of vertices is very large. Calculating suggestion scores from  $u$  to all vertices is infeasible. The selection method must satisfy two requirements: selected candidates should have high potentiality to establish connection from  $u$  and should be suitable for calculation method to ensure acceptable computation time.

Given  $C_u$  as candidate set of  $u$ ,  $L$  as maximum number of candidates and a threshold  $\mu$ , the candidate selection method sorts all vertices having path of length two to  $u$  in descending order by the number of mutual neighbours. Then, for each vertex  $v$  in this sorted list, it adds  $v$  to  $C_u$  if  $|C_u| < L$  and  $\mu \leq |N_u \cap N_v|$ .

In social networks, candidates are friends of friends of  $u$  but not friends of  $u$ . This comes from the fact that more than a half of new friendships in ZingMe and Facebook [12] are friends of friends. We sort candidates by number of mutual friends with assumption that users with more mutual friends with  $u$  may be easier to create new connections to  $u$ . The number of mutual friends and the number of candidates is also limited to improve prediction efficiency and to reduce computation time.

**Score Calculation**  $score_{u,v}$  indicates ability (i.e., the system's confidence level) to establish connection from  $u$  to  $v$ . It is calculated from two knowledge types: direct information between  $u$  and  $v$ ,  $aff_{u,v}$ , and indirect information from others on the graph,  $\psi_{u,v}$ . The computation  $score_{u,v}$  is defined of as follows:

$$score_{u,v} = w_d * aff_{u,v} + w_i * \psi_{u,v} \quad (3)$$

in which  $w_d$  and  $w_i$  are weights for direct and indirect information, indicating the level of their influence to the formation of the connection between  $u$  and  $v$ . Calculation method for  $\psi_{u,v}$  should leverage properties of connections of other vertices that can affect ability for  $u$  to connect to  $v$ . These properties for a pair of vertices include number and quality of paths between them such as length of a path or weight of each connection.  $\psi_{u,v}$  calculation can be turned into proximity measures in graph. Random walk with restart (RWR) is a well-known approach for this problem [15].

RWR can be defined as following equation:

$$r_i = (1 - \alpha) * r_i * A + \alpha * e \quad (4)$$

in which  $A$  is transition matrix,  $r_i$  is stationary distribution of RWR at iteration  $i$ ,  $e$  is unit vector with  $e_u = 1$  and  $\alpha$  is restart probability. A RWR starts at  $u$  can be seen as a particle starts at  $u$  and in each step, it moves to its neighbours with probability that is proportional to their weight and also returns to  $u$  with a probability  $\alpha$ . RWR stops after maximum  $m$  steps or when it converges with threshold  $\epsilon$ . If RWR stops at iteration  $i$ ,  $r = r_i$ .

We apply RWR as follows:

- For a target vertex  $u$ , build a local graph  $G_u = (V_u, E_u)$  in which  $V_u = \{u\} \cup N_u \cup C_u$  and  $E_u$  is the edges set containing those edges  $e_{v,w}, v, w \in V_u$
- Run RWR at  $u$  on  $G_u$
- When RWR stops, set  $\psi_{u,v} = r_v$

After calculating scores, the candidates are ranked in descending order by the scores.

## 4 Experiments and Discussions

### 4.1 The Used Datasets

Experiments are performed to evaluate effectiveness of the proposed friend suggestion approach compared to some other approaches: the one currently deployed in ZingMe, Adamic-Adar score, number of mutual friends, and plain Random Walk with Restarts. The data used in these experiments are obtained from the real data collected by ZingMe, including:

- The list of new friendships established in ZingMe from August 08, 2013 to October 02, 2013 that consists 3,540,624 users and 26,040,831 new friendships.
- List of friends, schools, groups, and companies of users appeared in new friendships above.

We note that many features missed in the collected data, such as the number of mutual IP addresses and the number of mutual interactions. This may make negative effects on suggestion efficiency of the proposed approach.

Users who have number of new friendships smaller than 5 are eliminated from dataset in order to keep more active users for evaluation. The set of remaining users is called  $U$ . Then, for each user  $u \in U$ , we remove from friend list of  $u$  users who are not in  $U$  or make friends with  $u$  in the period of data collection. Finally,  $U$  consists 928300 users and on average, each user has 90 friends and 25 new friendships. We consider  $U$ , refined lists of friends, and lists of schools and groups of each  $u \in U$  as the snapshot of the network that is used to calculate suggestions for tested users.

In order to evaluate performance of approaches to users with different numbers of friends, we randomly choose three user sets  $T20$ ,  $T50$ , and  $T100$  consisting of about 1000 users having number of friends from 20 to 30, from 50 to 60, and at least 100, respectively.

### 4.2 The Experiment Design

We divide new friendships into cross-validation set consisting ones from August 08, 2013 to August 27, 2013, and test set consisting ones from August 28, 2013 to October 02, 2013. We estimate parameters or weights of the proposed approach through the cross-validation set.

We could only collect data for three features: the number of mutual friends, the number of mutual schools and the number of mutual groups. Therefore, we set weights of other ones to 0. We change weights of those three features by step 0.1, keep weight for the number of mutual friends higher than the others and finally get the weights for them equal to 0.5, 0.3 and 0.2, respectively. Through estimation, we set  $w_d, w_i$  to 0.4 and 0.6.

For RWR,  $L = 10000$  helps RWR run in acceptable time from 1 to 2 seconds per users on average.  $\mu = 5$  outputs each  $u \in T20$  about 100 suggestions and much higher for users in  $T50$  and  $T100$ . We set  $c = 0.4$ ,  $\epsilon = 10^{-4}$  and  $m = 50$ .

We obtain parameters of the current approach from the real system and present them in Table 1. The approaches using Adamic-Adar score and number of mutual friends consider all friends of friends

of user  $u$  as candidates for  $u$ . The RWR approach uses parameters as in the proposed approach but it sets strength of all edges to 1.

Table 1: Values of parameters of current approach

(a) Values of $t$				(b) Values of $e$				
$t_1$	$t_2$	$t_3$	$t_4$		1	2	3	4
1.7	1.5	1.4	1.1	1	0	2	1.9	1.6
				2	2	0	1.8	1.7
				3	1.9	1.8	0	1.4
				4	1.6	1.7	1.4	0

**The Used Metrics.** We use *precision* and the Area under the ROC curve (AUC) as the metrics to evaluate the system’s performance on friend suggestion.

The precision score indicates how many suggestions for  $u$  would actually become friends of  $u$ . We measure precision at top 100 suggestions with thresholds from 1 to 100. The precision score at threshold  $k$  is defined as follows:

$$P@k = \frac{\text{number of actual friends at top } k}{k} \quad (5)$$

We use the average precision scores at each threshold from all tested users as the finally precision score.

The AUC score indicates the expectation that a true suggestion, i.e., a suggestion actually becomes a friend of  $u$ , is ranked before a false suggestion, i.e., a suggestion does not become a friend of  $u$ , and is calculated as follows [18]:

$$AUC = \frac{S_0 - n_0(n_0 + 1)/2}{n_0 n_1} \quad (6)$$

in which  $n_0$  and  $n_1$  are the numbers of true and false suggestions, respectively, and  $S_0 = \sum_i r_i$  where  $r_i$  is the rank of  $i^{th}$  true suggestion in the ranked list. The final AUC score is calculated by the average of AUC scores of all tested users. We only measure AUC for the proposed approach and RWR one, and measure whole suggestion lists from these approaches.

### 4.3 The Experimental Results and Discussions

Figure 2 shows precision curves of the approaches for different user sets. We can see that in all cases, the proposed approach outperforms the current one. On average, the improvement is about 98%, 28%, and 7% in  $T20$ ,  $T50$ , and  $T100$ , respectively.

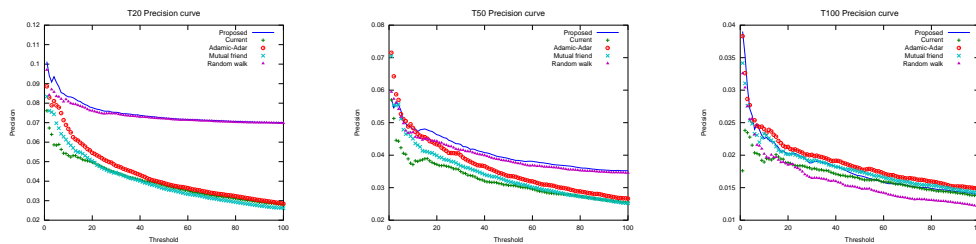
In  $T20$  and  $T50$ , the proposed approach also outperforms Adamic-Adar and number of mutual friends approaches with improvements, on average, about 85% in  $T20$  and 20% in  $T50$ . Its performance is slightly lower than those two approaches in  $T100$ .

Table 2 shows AUC scores of the proposed and plain RWR approaches. The proposed approach performs better than plain RWR one in both precision and AUC scores.

Our proposed approach outperforms other ones in almost all of cases. However, when the number of friends of target user  $u$  increases, the improvement tends to decrease.

This can be explained by missing features in datasets. The proposed approach uses more features than others but those features are not existed in datasets. They are the number of mutual IP addresses and the number of mutual interactions. Especially, when the number of friends of  $u$  increases, the number of mutual interactions between  $u$  and  $v$  may represent  $aff_{u,v}^f$  clearer than some other features under the hypothesis that the more number of mutual interactions two users have, the more affined they are.

Moreover, the increase in the number of friends often leads to drastic increase in the number of friends of friends of  $u$ . Then, with limitation  $L$  for the number of candidates, the proposed approach may not cover enough potential candidates. Meanwhile, all other approaches, except plain RWR one, calculate suggestion score for all friends of friends of  $u$ .



(a) The precision curve for  $T20$       (b) The precision curve for  $T50$       (c) The precision curve for  $T100$

Figure 2: The precision curves of the approaches for  $T20$ ,  $T50$ , and  $T100$  user sets with precision threshold from 1 to 100

Table 2: AUC scores of the proposed and the RWR approaches in all user sets

User set	Proposed	RWR
$T20$	<b>0.599</b>	0.589
$T50$	<b>0.620</b>	0.602
$T100$	<b>0.626</b>	0.582

By evaluating the approaches through different sets of users with different number of friends, we can see that the proposed approach performs better when users have smaller friends. Besides the reason from limitation  $L$ , another reason may be that increasing number of friends of users may reduce differences among candidates, i.e., differences among calculated affinities, and then cause the approach more difficulty to differentiate them. For example, with number of mutual friends feature only, if two candidates  $v_1$  and  $v_2$  have 5 and 6 mutual friends with  $u$ , respectively, difference between them is  $|0.5 * \log(6) - 0.5 * \log(7)| = 0.077$ . When number of their friends increase, number of mutual friends with  $u$  may increase. If they have 99 and 100 mutual friends with  $u$ , difference is  $|0.5 * \log(100) - 0.5 * \log(101)| = 0.005$ . This experimental result may indicate that the proposed approach should be used at the beginning use period of the networks or for those users with small number of friends.

## 5 Related Works

The link prediction problem for social networks was formally defined by Liben-Nowell and Kleinberg in [1]. They proposed and compared some approaches for this problem, such as shortest path, mutual neighbors, Jaccard, Adamic-Adar and so on. They evaluated on social networks of co-authorships via scientific papers. The effectiveness was evaluated over a random approach. These approaches only leverage network structure information.

There are many works on link prediction for social networks using only network structure information for prediction new connections. Huang [13] used information from clusters in the network. Rattigan and Jensen [2] focused on a smaller problem: anomalous link detection. They showed effectiveness of applying link prediction models for anomalous link detection problem. They also presented a fundamental problem causing low raw results of link prediction model. It is a massive class skew. Menon and Elkan [3] used matrix factorization and showed that their approach can combine more types of information. Sun et al. [4] exploited network structure to solve link prediction problem in heterogeneous networks. Not only appearance of connections, they also predicted time of the appearance. Some works [5], [6] solved this problem on growing networks instead of static networks.

One of reasons making most of works focus on network structure is difficulty in obtaining other kinds of information of users, such as attributes of users or their interactions. In real systems, these kinds of information can be obtained but user privacy should be considered. Machanavajjhala et al. [7] presented an approach for a compromise between suggestion accuracy and user privacy.

In addition to network structure, some works used network structure for the link prediction problem. Yin et al. [8] presented some factors that may effect on establishing connections and make it easier to connection to this user than that one. They showed that random walk was suitable for this problem and satisfied those factors. De et al. [9] hypothesized that connection occurrence between two nodes depended only on local properties. Scellato et al. [10] and Wang et al. [11] used location and mobility of nodes. For social networks that have these types of information, leveraging them may be a good direction.

Backstrom and Leskovec [12] exploited network structures, attributes of nodes and edges for this problem. They presented supervised random walk that guided random walk so that the random walker was more likely to visit a node that would connect to target node. But they did not use direct information between two nodes for prediction. Tylenda et al. [14] also used these types of information for growing networks. In addition, in their approach, adjacent vertices did not need to be connected. This may help choosing adjacent vertices more flexible.

RWR is a well-known technique for proximity measure on graph [15], [16], [17].

## 6 Conclusions and Future Works

With the goal of helping users of online social networks build and extend their relationships by suggesting to those users new friendships, we presented some formal definitions of the link prediction problem, the current approach of ZingMe and proposed a new approach for this social network. The proposed approach exploits knowledge sources from network structure, attributes of nodes and edges to archive direct information between two users and indirect information from other users in the network. These types of information are used to assess ability to establish friendships among users. Random walk with restart is the method used to calculate indirect type of information.

Some experiments were conducted to evaluate effectiveness of the proposed approach on a big dataset collected from ZingMe. The proposed, current and some other well-known approaches were evaluated on this dataset. The results from proposed approach have exceeded other ones. In some cases, a poor improvement of current approach over other ones can be explained by the missing of features in the dataset. We also suggest proposed approach for users who have small number of friends and who the proposed approach can output much better suggestion lists compared to other ones.

There are still many knowledge sources in ZingMe ignored, which may be useful for friend suggestion problem. For example, chat rooms a user has entered, games a user has played and so on. Collecting them and finding more sources should be considered.

Interest of users to suggestions may vary in different contexts. For example, when a user  $u$  visits profile page of a user  $v$ ,  $u$  may incline to make friends with friends of  $v$  at that time. So we can consider a context-aware recommendation approach.

There are many fixed parameters in the proposed approach, for example, the weights for the features in Equation 2, or  $w_d$  and  $w_i$  in Equation 3. A model to learn them may improve performance.

User feedbacks to presented suggestions can be a good knowledge source to remove bad suggestions and to avoid repeating suggestions to users. For instance, if a user  $v$  is suggested to  $u$  many times but  $u$  has not made friend with  $v$ , perhaps  $v$  is not a good suggestion for  $u$  although  $score_{u,v}$  might be high.

## References

- [1] David Liben-Nowell, Jon Kleinberg, The Link Prediction Problem for Social Networks, *Proceedings of the twelfth international conference on Information and knowledge management*. 556-559, ACM New York, NY, USA, 2003.



- [2] Matthew J. Rattigan, David Jensen, The case for anomalous link detection, *MRDM '05 Proceedings of the 4th international workshop on Multi-relational mining*. 69 - 74, ACM New York, NY, USA, 2005.
- [3] Aditya Krishna Menon, Charles Elkan, Link prediction via matrix factorization, *ECML PKDD'11 Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases*. 437 - 452, Springer-Verlag Berlin, Heidelberg, 2011.
- [4] Yizhou Sun, Jiawei Han, Charu C. Aggarwal, Nitesh V. Chawla, When will it happen?: relationship prediction in heterogeneous information networks, *WSDM '12 Proceedings of the fifth ACM international conference on Web search and data mining*. 663 - 672, ACM New York, NY, USA, 2012.
- [5] Alan Mislove, Hema Swetha Koppula, Krishna P. Gummadi, Peter Druschel, Bobby Bhattacharjee, Growth of the Flickr Social Network, *WOSN '08 Proceedings of the first workshop on Online social networks*. 25 -30, ACM New York, NY, USA, 2008.
- [6] Joshua O'Madadhain, Jon Hutchins, Padhraic Smyth, Prediction and Ranking Algorithms for Event-Based Network Data, *ACM SIGKDD Explorations Newsletter, Volume 7 Issue 2*. 23-30, ACM New York, NY, USA, 2005.
- [7] Ashwin Machanavajjhala, Aleksandra Korolova, Atish Das Sarma, Personalized Social Recommendations: Accurate or Private?, *Proceedings of the VLDB Endowment, Volume 4 Issue 7*. 440 - 450, VLDB Endowment, April 2011.
- [8] Zhijun Yin, Manish Gupta, Tim Weninger, Jiawei Han, A Unified Framework for Link Recommendation Using Random Walks, *ASONAM Proceedings of International Conference on Advances in Social Network Analysis and Mining*. 152 - 159, 2010.
- [9] Abir De, Maunendra Sankar Desarkar, Niloy Ganguly, Pabitra Mitra, Local Learning of Item Dissimilarity Using Content and Link Structure, *RecSys '12 Proceedings of the sixth ACM conference on Recommender systems*. 221 - 224, ACM New York, NY, USA, 2012.
- [10] Salvatore Scellato, Anastasios Noulas, Cecilia Mascolo, Exploiting Place Features in Link Prediction on Location-based Social Networks, *Proceedings of the 12th international conference on Information and knowledge management*. 1046-1054, ACM New York, NY, USA, 2011.
- [11] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, Albert-Lszl Barabasi, Human Mobility, Social Ties, and Link Prediction, *KDD '11 Proceedings of the 17th ACM international conference on Knowledge discovery and data mining*. 1100-1108, ACM New York, NY, USA, 2011.
- [12] Lars Backstrom, Jure Leskovec, Supervised random walks: predicting and recommending links in social networks, *WSDM'11 Proceedings of the fourth ACM international conference on Web search and data mining*. 635 - 644, ACM New York, NY, USA, 2011.
- [13] Zan Huang, Link Prediction Based on Graph Topology: The Predictive Value of the Generalized Clustering Coefficient, *Proceedings of KDD '06 workshop on Link Analysis: Dynamics and Static of Large Networks*. 2006.
- [14] Tomasz Tyenda, Ralitsa Angelova, Srikanta Bedathur, Towards Time-aware Link Prediction in Evolving Social Networks, *SNA-KDD '09 Proceedings of the 3rd Workshop on Social Network Mining and Analysis*. Article No. 9, ACM New York, NY, USA, 2009.
- [15] Hanghang Tong, Christos Faloutsos, Jia-Yu Pan, Fast Random Walk with Restart and Its Applications, *ICDM '06 Proceedings of the Sixth International Conference on Data Mining*. 613-622, IEEE Computer Society Washington, DC, USA, 2006.
- [16] Purnamrita Sarkar, Andrew W. Moore, Fast dynamic reranking in large graphs, *WWW '09 Proceedings of the 18th international conference on World wide web*. 31-40, ACM New York, NY, USA, 2009.
- [17] Hanghang Tong, Yehuda Koren, Christos Faloutsos, Fast Direction-Aware Proximity for Graph Mining, *KDD '07 Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 747-756, ACM New York, NY, USA, 2007.
- [18] Charles X. Ling, Jin Huang, Harry Zhang, AUC: a better measure than accuracy in comparing learning algorithms, *AI'03 Proceedings of the 16th Canadian society for computational studies of intelligence conference on Advances in artificial intelligence*. 329-341, Springer-Verlag Berlin, Heidelberg, 2003.