A Method of User Recommendation in Social Networks Based on Trust Relationship and Topic Similarity*

Yufeng Ma, Zidan Yu, and Jun Ding

Department of Computer Science and Engineering
East China University of Science and Technology
Shanghai, 200237, China
045120100@mail.ecust.edu.cn

Abstract. In the research area of user recommendation in social network sites (SNS), there exist problems that some algorithms based on the structure of SNS are resulting in low quality recommendation results due to lack of model and mechanism to express users' topic similarity, some algorithms which use topic model to measure the theme similarity between users cost a lot of time because of the topic model have a high time complexity in case of large amounts of data. This paper proposed a hybrid method for user recommendation based on trust relationship and topic similarity between users, aiming to widening their circle of friends and enhancing user stickiness of SNS. Two main steps are involved in this process: (1) a trust-propagation based community detection method is proposed to model the users' social relationship; (2) a topic model is applied to retrieve users' topics from their microblogging, and gain the recommendations by the topic similarity. Our research brings two major contributions to the research community: (1) a Peer-to-Peer trust model, PGP, is introduced to the field of community detection and we improve the PGP model to compute trust value more precise; (2) a distributed implementation of the topic model is proposed to reduce total execution time. Finally, we conduct experiments with Sina-microblog datasets, which shows the model we proposed can availably compute the trust degree between users, and gain a better result of recommendation. Our evaluation demonstrates the effectiveness, efficiency, and scalability of the proposed method.

Index Terms: trust degree, community detection, topic model, topic similarity, user recommendation.

1 Introduction

As the development of Social Network Site, a very large scale of relationship graph is formed between users. In recent years, one of the most popular social network sites should be Sina microblog. At the end of 2013, Sina microblog has nearly 600 million registered users, 60 million daily active users, and 200 million daily microblogs. As for the reasean why SNS can ensure users to be active, it is not noly because they can maintain the real friend relationship, but also because they can find more attentive users and expand their social circle. Therefore, user recommendation plays an important role in SNS, its effect has a direct influence of the popularity of social networking sites.

* This work is supported by the science and technology support program (2013BAH11F03).

H.-Y. Huang et al. (Eds.): SMP 2014, CCIS 489, pp. 240–251, 2014.
© Springer-Verlag Berlin Heidelberg 2014

Much of the recent researchs of social network sites' user recommendation mainly focuse on two fileds: one is based on users' characteristics, and the other is based on users' relationship graph. User recommendation methods which based on user characteristics can fall into three categories: content based, common topics based, user labels based. [1] used personal informations of users as the content and calculated similarity for user recommendation; Piao S [2] applied term vectors extracted from users' tweets to represent user's topics, then recommended friends with similar topics; Gou L [3] used the association rules to compute the label similarity between users for recommendation.

However, the algorithms based on users' characteristics can only recommend similar friends and the recommendation is very tedious for users. [4] built a graph by the following relationship between users and proposed a link prediction method for recommendation based on the structure similarity. [5] utilized the adjacency relationships between users to calculate the similarity matrix, then recommended friends by relevant information about the users' network topology. The method based on users' relationship graph always recommend familiar users, and it will not be unable to recommend potential friends with the same topics accurately.

At the same time, the recent research on friend recommendation did not make full use of the trust relationship between users, a closest friend with a high trust degree did not play a proper role for recommendation in these methods because closest friends were considered to have the same effect as unfamiliar friends.

PGP (Pretty Good Privacy) is a model which uses the asymmetric encryption to protect data security. Unlike the trust model those use CA (Certificate Authority) as a trusted third party, PGP leaves the trust of the initiative to users and utilizes the recommender trust model to measure the trust level between users. However, there are two obvious drawbacks when PGP is applied in large-scale social networks:

1. The trust chain can not include nodes more than two, that means it only has two types of edge in the trust chain: one of this is direct trust and the other is recommender trust. It is obviously not enough for large-scale social network
2. PGP gives only three or four trust grades, so it is difficult to distinguish the trust degree between users precisely.

For these problems, we propose a hybrid method for user recommendation based on the trust degree and topic similarity between users. Firstly, we present an improved PGP model, the trust propagation model (**TPM**), to calculate the trust-degree matrix between users, and a trust-degree based community detection method is established. Secondly, we pick out the community which has the target user, and retrieve all users' topics from their microblogging by the topic model in this community, then recommend friends to the target user by the topic similarity.

2 Whole Framework

The algorithm in this paper aims at recommending potential friends for the SNS users. We realize there are two kinds of connections between users in the SNS through the previous analysis, one is the link in networks between users, and the other is users' topic similarity in all kinds of topics. The goal of our algorithm is to combine these two kinds of connections. The overview of our algorithms is shown in Figure 1.

Firstly we divide users into groups by the community detection algorithm, then extract the user's topic distribution in the community and calculat the topic similarity between the target user and all other users in this community, finally give the user's personalized friends recommendation. There is a brief introduction to these steps..

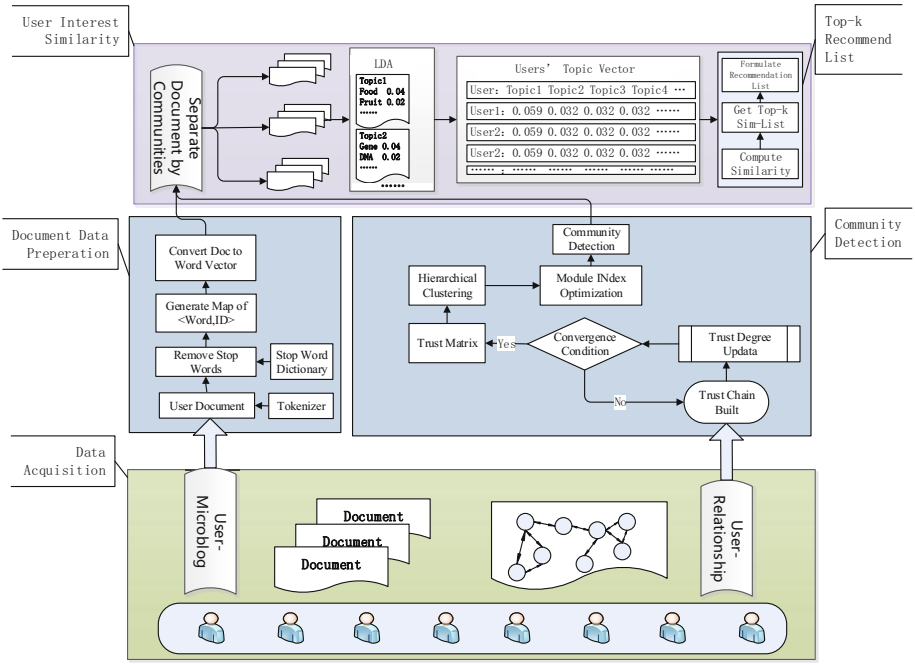

Fig. 1. Framework overview

1. Community discovery algorithm

For the existing user relationship, based on the PGP trust model we propose a modified method, to measure the trust between nodes, and then use fast hierarchical clustering algorithm which proposed by J. H. Ward [6] , with the aim to optimize module of the whole social network degree, to generate the final community. The specific calculation process is in section 3.

2. Users' topic similarity calculation

After the community division, the network topology relationship of the users in the community become more compact, then a user's microblogs can be seen as one document, This document contains the topic of the user's attention, and we adopt LDA model which proposed by Blei.D.M [7] to calculate each document topic distribution, then calculate the topic similarity of each document. The specific calculation process is in section 4.

3. Generate friend recommended list

After user topic similarity calculation for recommended users, apart from its users, select the Top-K greatest similarity users, to generate recommendation list. At this point, we have completed the whole process of personalized friends' recommendation.

3 The Community Detection Algorithm Based on User Trust Chain

Considering the actual situation of the social network, people give more trust to their close friends, and a group of users, in which people have high trust degree between each other, will form a community. Referencing the presenter trust models in the PGP, this paper propose a modified community discovery algorithm based on trust-chain, which can overcome the two drawbacks of PGP: trust chain maximum length is two, and trust measure size is not enough.

3.1 The Definition of Community in SNS

To build a social network un-weighted digraph $G = (V, E)$, V denotes vertex (user) collection, $|V|=n$, E denotes the collection of user relationship, e_{ij} denotes the edge linked v_i, v_j. Such a social network graph can also be denoted as the adjacency matrix $A=a_{ij}(v_i,v_j \in V)$, when $(v_i,v_j) \in E$, $a_{ij}=1$.

Divided G into k, get a partition which has k vertex collections, $\varphi = \{N_1, N_2 \ldots N_k\}$. If $N_i \in \varphi$, all the nodes' trust chain in N_i are intensive, and all the nodes' trust chain out of N_i are sparse, then φ is the community partition of G which based trust chain.

3.2 The Calculation of Trust in Trust Chain

If there is an edge between v_i and v_j, it donates user i has followed user j, it also donates i has a trust with j. If the total trust of each user is 1, and he equally assigns it to all the users he has followed. Initialize the trust:

If i has followed j, and i has followed $\sum_{k=1}^{n} a_{ik}$ users, then the initialized trust of i to j is:

$$Tru(i, j) = \frac{1}{\sum_{k=1}^{n} a_{ik}} \tag{1}$$

Assuming that trust can be spread in two ways: series and parallel:

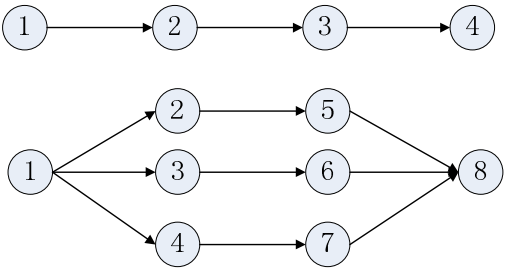

Fig. 2. The propagation of trust degree

For the series of trust propagation, the trust of each path will decrease with the increase of the intermediate node,

$$Tru(1,4) = Tru(1,2) \times Tru(2,3) \times Tru(3,4)$$

For parallel transmission of trust, the trust will increase with the increase of the path,

$$Tru(1,8) = sum\begin{Bmatrix} Tru(1,2) \times Tru(2,5) \times Tru(5,8) \\ Tru(1,3) \times Tru(3,6) \times Tru(6,8) \\ Tru(1,4) \times Tru(4,7) \times Tru(7,8) \end{Bmatrix}$$

Combined with the two assumptions, given the following method to update the trust between i,j:

1. Given the depth of the threshold value, through the breadth-first search, get m trust chains from i to j.
2. For these m trust chains, the intermediate nodes of each trust chain is v` = {v1`, v2`…vk`} , v1` denotes i, vk` denotes j. Then the collection of trust TChain = {Tru(v1`,v2`),Tru(v2`,v3`)…Tru(vk-1`,vk`), $TChain_q^p$ denotes the q trust in trust chain p.
3. Add all of the m trust chains, then the trust between i and j is:

$$Tru(i, j) = \sum_{p=1}^{m} \prod_{q=1}^{k} TChain_q^p \tag{2}$$

Then give the following iteration method, to calculate trust between any two nodes:

GET THE TRUST-MATRIX

1 **for** $(i,j) \in \{(v_i, v_j) | v_i, v_j \in V, v_i \neq v_j\}$
2 **do** initialize $tru(i,j)$ acc. to Eq.1
3 times $\leftarrow 1, LT \leftarrow limit\ Times$
4 $\text{tru}' = 0$
5 **while** $tru \neq tru'$ or $0 < times < LT$
6 **do** $tru'(i,j) = update(tru(i,j))$
acc. to Eq.2
7 return tru

Fig. 3. The method of trust matrix calculation

So we get the trust matrix to describe any two nodes.

Similarly, we consider the trust between node *i* and community *C*, it defined as the average trust of node *i* and all the nodes in community *C*:

$$Tru(i,C) = \frac{1}{|C|}\sum_{j \in C} Tru(v_i, v_j) \tag{3}$$

|C| denotes the nodes' number of community *C*. The trust between community and community, defined as trust root mean square average of any two nodes in the community :

$$Tru(C_1, C_2) = \sqrt{\frac{\sum_{i \in C_1}\sum_{j \in C_2} Tru^2(v_i, v_j)}{|C_1|+|C_2|}} \tag{4}$$

3.3 The Steps of Hierarchical Clustering

In the last section, this paper puts forward how to calculate the trust between two nodes. The next question is how to cluster these nodes to form the community. Here, we use the bottom-up hierarchical clustering algorithm which proposed by J. H. Ward [6], the specific algorithm is as follows:

Divide all the nodes of G, get community partition $\varphi = \{\{v\}, v \in V\}$, each community has only one node. Calculate all adjacent nodes trust first, and then through the following step iteration to combine the community:

```
HIERARCHICAL CLUSTERING FOR COMMUNITY
1  TRU = {tru(C_i, C_j)|C_i, C_j ∈ φ, i ≠ j}
2  while size(φ) ≠ 1
3        do Choose {(C_1, C_2)|tru(C_1, C_2) = max(TRU)}
4           C_3 ← C_1 ∪ C_2
5           φ ← (φ − {C_1, C_2}) ∪ C_3
6           TRU ← {tru(C_i, C_j)|C_i, C_j ∈ φ, i ≠ j}
```

Fig. 4. The hierarchical clustering procedures of community

After the process above, to get the final community partition $\varphi_n = \{V\}$. This is a process of hierarchical clustering, according to the order of nodes are merged into the community, dendrogram can be constructed. The leaves of the dendrogram is all the vertices on the social network graph, and the internal nodes of the dendrogram corresponds to the "merge" steps of the algorithm: just that corresponds to a new community which merges its two children nodes.

3.4 Module Index

Module index *Q* [8] is used to depict community features of strength. Its main idea is based on "network community structure, the more obvious, the greater the difference between it and the random network". In general, a higher degree of module of network represent the partitioning effect is better. For the above given hierarchy clustering figure, consider some form of division, graph will be divided into *k* communities. *m* denotes the total number of network connections, m_i denotes the

number of network connections in C_i, d_i denotes the sum of the node degrees in community i. The definition of Q function is the difference between the actual number of connections and expectation connections in randomly connected network:

$$Q = \sum_{i=1}^{k} \left[\frac{m_i}{m} - (\frac{d_i}{2m})^2 \right] \tag{5}$$

If Q tends to be 1, it donates the community has the very strong structure. In the actual network, the value is generally between 0.3 ~ 0.7. Finally using the hierarchical clustering results, according to the maximum module degrees of cutting, then get the community partition.

4 Extract User Topic Distribution and Similarity Calculation

On the microblogging platform, the main performance way of users interaction is the microblog contents. Microblog contents host the user's will, goals, and even social relationship. How to use topic learning and unsupervised clustering, according to these microblog contents, get the user's hobby then clustering the users who have similar microblog contents is a core problem in social network data mining field. In topic models, the most common is LDA (Latent Dirichlet Allocation) model.

Because each microblog is short, in order to extract users' topics accurately and rapidly, this paper will treat each user's all microblogs as a document. Then we use LDA model to extract the topics of these documents, to get each user corresponding topic distribution, ultimately through the cosine similarity calculation, getting the topic of the similarity between the users, the greater the value shows that the better correlation between the two users.

4.1 Latent Dirichlet Allocation Topic Model

LDA (Latent Dirichlet Allocation) is a three layers of Bayesian probability model. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. It assumes there are D document and K topics in the corpus. For each document d in D, there are N_d words. These K topics are shared by all the documents, but for each document has a corresponding topic distribution. For each topic has a corresponding word distribution. LDA assumes the following generative progress for each document d in a corpus D:

CREAT A DOCUMENT

1 **for** each topic $k \in \{1...k\}$
2 **do** draw $\varphi \sim Dir(\beta)$
3 **for** each document $d \in \{1...d\}$
4 **do** draw $\theta_d \sim Dir(\alpha)$
5 **for** each word $w \in \{1...N_d\}$
6 **do** draw $\xi_{d,n} \sim Mult(\theta_d)$
7 draw $W_{d,n} \sim Mult(\psi_{\xi_{d,n}})$

Fig. 5. The generative process of a document

Figure 6 shows the probabilistic graphical modeled LDA:

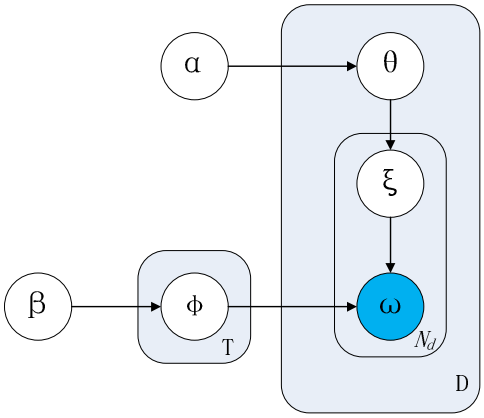

Fig. 6. The probabilistic graphical model of LDA

As the figure makes clear, there are three level to the LDA representation. The parameters α and β are corpus-level parameters, assumed to be sampled once in the process of generating a corpus. The variables θ_d are document-level variables, sampled once per document. Finally, the variables ζ and ω are word-level variables and are sampled once for each word in each document.

The topic distribution θ_d of document d is sampled from Dirichlet distribution. The word distribution φ_ζ of ζ topic is sampled from Dirichlet distribution. Both of the Dirichlet distributions are independent. θ for example, its corresponding distribution function is as follows:

$$p(\theta \mid \alpha) = \frac{\Gamma(\sum_{i=1}^{T} \alpha_i)}{\prod_{i=1}^{T} \Gamma(\alpha_i)} \prod_{t=1}^{T} \theta_t^{\alpha_t - 1} \tag{6}$$

T denotes the number of topics, α_i (i∈[1,T]) is each component of α, Γ is the gamma function.

We obtain the probability of a corpus:

$$p(D \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) p(w_{dn} \mid z_{dn}, \beta)) d\theta_d \tag{7}$$

Now require the distributions of ϕ and θ. Here we use Gibbs extraction, it is easy to implement and can effectively extract topic from the large-scale corpus. It's the most popular LDA model extraction algorithm. Gibbs extraction iterates sampling the topic of words in each position instead of directly computing ϕ ζ and θd. Once the topic of words in each position is definite, ϕ ζ and θd can be calculated. Then we can get the topic distribution of documents, and the word distribution of topics.

4.2 Topic Similarity Calculation

Assuming that from LDA model, we can get the topic distribution $Topic_A$、$Topic_B$ of user A and B. Then we use Cosine Similarity to calculate the topic similarity between A and B.

$$sim(A, B) = \frac{Topic_A \bullet Topic_B}{|Topic_A| \times |Topic_B|} \tag{8}$$

If Cosine similarity is close to 1, it means the more similar the two topic of the user's attention.

5 Experimental Analysis

5.1 Experimental Data and Parameter Settings

Tang Jie, the professor in Tsinghua University, has crawled 1.7 million users and 0.4 billion following relationships among them [9]. In this paper, we use following method to extract instances: 1. five adjacent user nodes are selected as the seeds; 2. In each iteration, the depth priority method is used to fetch the nodes which far away from the current user; the breadth first method is used to fetch the adjacent nodes of current users. Finally, we obtain a social network with 1538 nodes, and 732643 relationships.

This paper consider the maximum length of trust chain is 4 due to the complexity of the algorithm. When calculate the distribution of the users documents, we use ICTCLAS as the tokenizer, and get 65023 words totally. LDA is conditioned on three parameters. In this paper, they are set as T=30, $\alpha = 50/T$ and $\beta = 0.01$ according to [10].

5.2 Evaluation Method

In the experiment, we recommend *Top-K* friends for users in the offline data set by the algorithm posted in this paper. In order to verify the test results, this paper adopts the following recommendation system evaluation standards: *precision* rate, *recall* rate. Let $R(u)$ denotes the list of users that are recommended to u, and $T(u)$ denotes the list of friends of u. The *precision* and *recall* are defined as follows:

$$precision = \frac{\sum_{u \in U} |R(u) \bigcap T(u)|}{\sum_{u \in U} |R(u)|} \tag{9}$$

$$recall = \frac{\sum_{u \in U} |R(u) \bigcap T(u)|}{\sum_{u \in U} |T(u)|} \tag{10}$$

F-measure is the harmonic mean of *precision* and *recall:*

$$F = \frac{2 \times precesion * recall}{precesion + recall} \tag{11}$$

5.3 Experimental Results and Analysis

Choose simple based on link relations and simple based on the topic method as a benchmark experiment. In order to get the final result, the algorithm proposed in this paper will compare with two kinds of benchmark experiments. Table 1 show the results, R denotes *Recall* rate, P denotes *Precision* rate, *F* denotes harmonic average.

According the results of table 1, we can get several evaluation index charts of the algorithm, as shown in figure[7-9]:

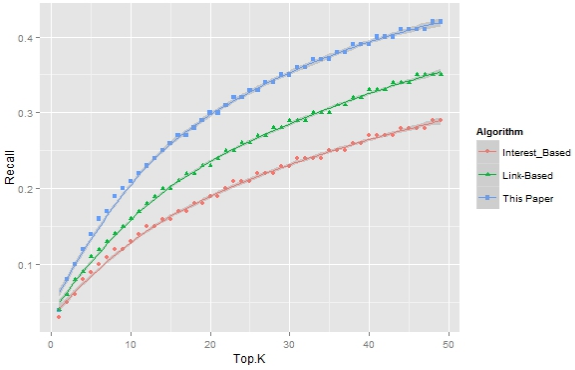

Fig. 7. Comparing the recall

Table 1. The comparison result of different algorithm

Top-K	This Paper			Link-Based			Topic-Based		
	R	P	F	R	P	F	R	P	F
5	9.01%	25.43%	0.090	6.22%	14.82%	0.062	4.12%	7.86%	0.041
10	10.46%	20.67%	0.105	6.65%	11.25%	0.067	4.18%	6.12%	0.042
15	10.52%	17.75%	0.105	6.31%	9.15%	0.063	3.91%	5.13%	0.039
20	10.32%	15.78%	0.103	5.91%	7.92%	0.059	3.64%	4.51%	0.036
25	9.96%	14.33%	0.100	5.49%	6.94%	0.055	3.40%	4.05%	0.034
30	9.61%	13.22%	0.096	5.12%	6.24%	0.051	3.19%	3.70%	0.032
35	9.26%	12.31%	0.093	4.78%	5.68%	0.048	3.00%	3.42%	0.030
40	8.97%	11.62%	0.090	4.50%	5.22%	0.045	2.85%	3.19%	0.028
45	8.64%	10.96%	0.086	4.22%	4.82%	0.042	2.69%	2.98%	0.027
50	8.42%	10.53%	0.084	4.03%	4.55%	0.040	2.59%	2.85%	0.026

As the figures shown, both precision rate and recall rate of this algorithm is significantly better than the benchmark algorithms, The reason is that in this paper, we consider both the links between users and topic information of the user's interests, which is better than to consider only a single aspect of the informations.

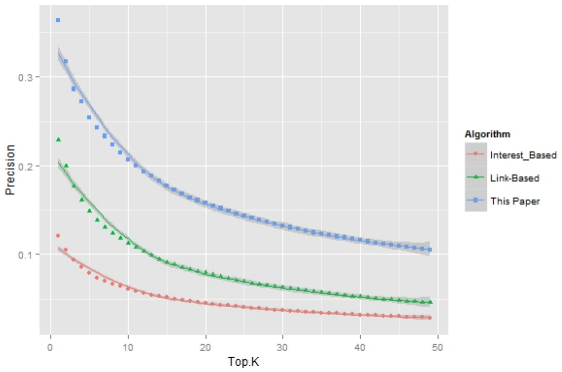

Fig. 8. Comparing the precision

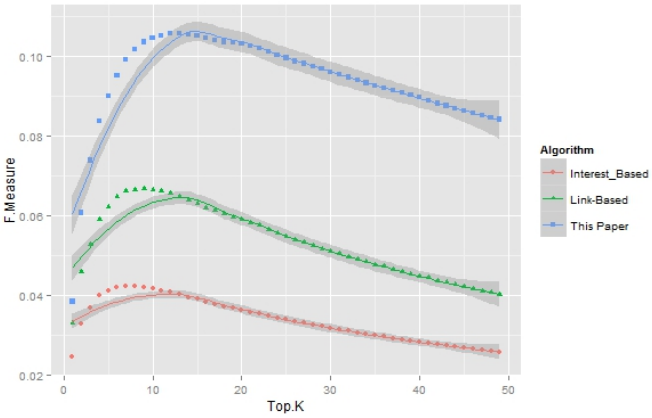

Fig. 9. Comparing the F value

6 Conclusion

This paper presents a novel method to recommend users in social networks, in which both link-based similarity and topic-based similarity are considered. The method exploits a two-phase process to provide the trust based and topic sensitive recommendations for social network users. Firstly, we utilize the trust recommend models in PGP to model the trust relations between the user nodes, compute the trust degree between each pair of users, divided the whole social network into several sub-community networks based on the trust degree. Secondly, we use LDA to model the topics of each users in a community and analyze the topic similarity between users, finally, we recommend the potential friends by the rank of these similarity. Experimental results on real-world data demonstrate that the proposed method outperforms other algorithms in terms of precision, recall, and F-measure. In our

future work, we plan to deploy our method onto the real social network site. This will allow us to collect users' feedbacks on our results, which can be helpful for us to adjust the recommendation strategies.

References

[1] Jeckmans, A., Tang, Q., Hartel, P.: Privacy-Preserving Profile Similarity Computation in Online Social Networks. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 793–796. ACM Press, New York (2011)

[2] Piao, S., Whittle, J.: A feasibility study on extracting twitter users' topics using NLP tools for serendipitous connections. In: 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust (passat) and 2011 IEEE Third International Conference on Social Computing (socialcom), pp. 910–915. IEEE (2011)

[3] Gou, L., You, F., Guo, J., et al.: SFViz: topic-based friends exploration and recommendation in social networks. In: Proceedings of the 2011 Visual Information Communication-International Symposium, p. 15. ACM (2011)

[4] Yin, D., Hong, L., Xiong, X., et al.: Link formation analysis in microblogs. In: The 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011), Beijing, China, pp. 1235–1236 (July 2011)

[5] Armentano, M., Godoy, D., Amandi, A.: Recommending Information Sources to Information Seekers in Twitter. In: Proceedings of the IJCAI: International Workshop on Social Web Mining, Barcelona, Spain (2011)

[6] Ward Jr., J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58(301), 236–244 (1963)

[7] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)

[8] Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69(2), 026113 (2004)

[9] Zhang, J., Liu, B., Tang, J., et al.: Social influence locality for modeling retweeting behaviors. In: IJCAI 2013 (2013)

[10] Weng, J., Lim, E.P., Jiang, J., et al.: Twitterr rank: finding topic sensitive influential twetterers. In: Proc. of the 3rd ACM International Conference on Web Search and Data Mining (2010)