Location Based Friend Recommendation for Online Social Network with Hypercube

J.Q. Zhu, M. Zheng
School of Computer and Information Engineering
Tianjin Normal University
Tianjin, China

***Abstract*—Nowadays, friend recommendation is one of the most popular applications for online social network, but it is often difficult to find truly useful information by using this function. The inaccurate friend recommendation results largely block the online social interaction among micro-blog users. Thus, in this paper, we propose to combine the user's interest and the user's location together. By recommending surrounding users who have similar interest to each other, micro-blog users are provided an effective way of interacting with surrounding people that they may want to know. Simulation results demonstrate the advantage of our proposed algorithms compared with the traditional friend recommendation approaches.**

Keywords-social networking; recommendation; hypercube

I. INTRODUCTION

Micro-blogging systems, especially Twitter and Weibo, have become extremely popular nowadays. For example, Twitter has more than 140 million active users and over 340 million messages posted per day[1]. Weibo has also accumulated over 300 million users and more than 1,000 Chinese tweets are being posted per second [2]. As an essential factor for users to find truly useful information in micro-blogging systems with huge resources, friend recommendation has drawn much research efforts. However, previous works on friend recommendation seldom consider user interest feature in friend finding. Zuo et al. [3] found that people with similar characteristics are more likely to form ties with each other. The authors in [4] also demonstrated that the contact between similar people occurs at a higher rate than among dissimilar people. Therefore, we believe searching for interest match people can enhance both the user' online and offline social interactions.

In this paper, we propose a location based friend recommendation method to recommend surrounding people with similar interest to each other. Our method consists of two main parts. Firstly, we generate the user interest vector to mine the user's interest from tweet contents. Then, we try to measure users' interest similarity with multiple interest topics under a binary cube structure. Matched users are ranked according to their topic distance in the binary cube. Experimental results prove that our algorithms achieve high performance in interest match.

II. CONSTRUCT USER INTEREST MODEL

The match of users' interest is an intuitively important feature. When recommending a user to another user, two users with similar interest are more likely to talk to each other. To calculate the match, the largest challenge is to generate the user interest model on micro-blogging systems, which differs from traditional user interest models because contents in the tweets are short, contain a wide variety of topics, as well as even noisy. Based on the previous studies, topic modeling clustering algorithm like LDA [5] is usually used to predict the topic distribution of a document. In the standard LDA, a document contains a mixture of topics, represented by a topic distribution, and each word has a hidden topic model. While this is true for long documents, for short micro-blog tweets, a single post is most likely to be about a single topic [6]. To achieve this, we introduce to modify the LDA algorithm to obtain the latent topics from micro-blog tweets contents.

To begin with, we crawl a user's recent tweets from sinaWeibo, which is a twitter like micro-blogging system in China. For a user *i*, we define *Di* as the set of recent tweets he posted. In our work, we assume *Di* is user *i*'s 1,000 most recent tweets. For *Di*, after word segment by *jieba* tool(atool used for Chinese word split), we eliminate all the stop words in the posted tweets, only keeping a word if it is identified as a noun. Then, LDA model is used to assign these tweets to 10 predefined topics, which is an empirical value of interest categories. In LDA calculating, a tweet is assigned to multiple topics and its relevance to each topic is represented by $p(c|Ti)$, where *c* denotes the tweet and *Ti* is a specific topic .Among all these topics, we let the topic which the maximum value of $p(c|Ti)(i=1,2,3\dots10)$ indicate its hidden topic. Thus, each tweet is assigned a single topic with a high probability.

Finally, by LDA calculating, each micro-blog user gets atopic vector, which shows his interest degree to each topic. An example of the topic vector is shown below, from which we acquire the user that vector W attached to prefers sports and travel to other interest topics.

Example: Topics (Entertainment, Sports, Health, Travel, Education, Culture, Military, Technology, Vehicle, Financial)

W=(0 ,0.21 , 0.06 , 0.30 , 0.05 , 0.11 , 0.03 , 0.02 , 0.09 ,0.13)

III. BINARY HYPERCUBE MODEL

A. Binary Hypercube Construction

Generally, if there are three or more than three interest topics, we can describe the difference of various users by using a multiple dimensional space. Each topic of the user is represented as a component of the vertex's coordinate in hyper space.

© 2015. The authors - Published by Atlantis Press

To achieve high matching performance, we propose a binary cube structure (binary T-space) in this section. The binary hypercube is a special cube in which each topic has a binary value: 0 or1. To construct a binary cube, we should change users' interest vectors into binary vectors at first. The dimensional values in a binary vector are either 0 or 1, where value 1 denotes the corresponding dimensional topic is a key topic, while value 0 represents the direct opposite of value 1. In order to transform interest topic vectors to binary vectors, two steps are taken as follows. First, key topics should be selected from the interest vector. Here the key topics represent the small subset of topics which the user enjoys the most among the ten topics. This is done by transforming component values of the user interest vectors less than a threshold γ to zero and keeping the component values if and only if they are not smaller than *γ*. Note that the component of the users' topic vector we acquired by the modified LDA calculating in our experiment have two features: 1) Most of the components have values less than0.1; 2) Only a few component values are larger than 0.1. Thus, we let the threshold γ be 0.1 in our simulation. Then, we change the component value of each key topic to1, while letting the values of other topics be 0. In this way, the interest vector is transformed into the binary vector. For instance, W=(0 ,0.21 , 0.06 , 0.30 , 0.05 , 0.11 , 0.03 , 0.02 , 0.09 ,0.13) is transformed to the binary vector W*= (0, 1, 0, 1, 0,1, 0, 0, 0, 1) by taking these steps.

After mapping binary vectors into the hypercube, a 10-dimensional binary cube is formed. More specifically, at each dimension in the binary cube, users are separated based on whether they have the same interest topics or not. Moreover, users have identical key topics are mapped into the same node in the cube, even though each topic may have many different values. Figure 1 shows a concrete 4-D binary cube constructed based on four different interest topics. In figure 1, we assume a group of users are mapped into a total number of 16 nodes in the cube. Node 0000 represents the set of users who are interested in none of these four topics.

B. Match Detection

We use topic distance to measure interest similarity between two users in the binary T-Space. The topic distance between user *i* and user *j* is set to *D(ij)*, which is the hamming distance between users *i* and *j*. In figure 1, we assume user Ais the source user. To match interest similarity for user A who mapped to node 0100, we perform the following processes:

1). We rank all the users (except A) that mapped into node0100 to similarity level 1, since these users have the same key topics with user A.

2). Other users are ranked to different similarity levels according to their topic distances to user A.

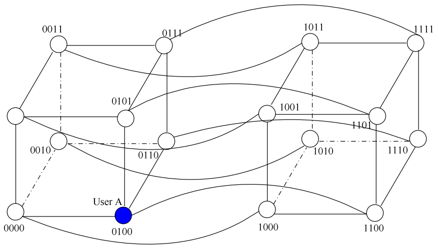

FIGURE I. A BINARY HYPERCUBE WITH FOUR TOPICS

IV. SHORTCUT BASED RECOMMENDATION

We now consider a new issue of our friend recommendation. It is obvious that there exist some inactive micro-blog users who post messages and update their micro-blog with a low frequency. For instance, user *i* does not get used to using micro-blog. He only logs in his micro-blog and posts a tweet once a month. Thus, user interest vectors we generated may not precisely reflect the profile of users. In[7], the authors depicted a person's interest can impact his friends' interest in the physical world, and vice versa. Work[8] proposed offline friends usually have potential hobbies in common. Therefore, in terms of a more precise friend recommendation, we propose to use offline friendship for more extensive user interest discovery.

Here, we propose the idea of shortcut. Traditionally, in our binary T-space, two users are connected if they differ in one topic. When two users are more than one topic distance away, there will be no direct connection between them. Here, weal low a controlled jump to a micro-blog user that is always more than one topic distance away in the binary cube. Such a controlled jump is called a shortcut. As shown in figure 2, the path from node 0000 to node 0111 is then a shortcut.

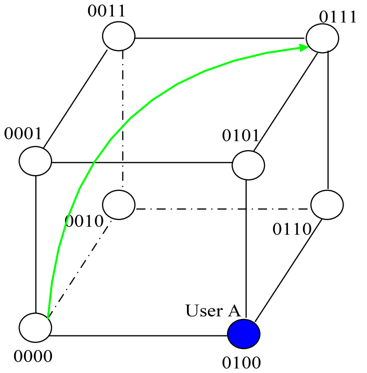

FIGURE II. A SHORTCUT IN THE CUBE

Intuitively, given the friend information of a user, we can see that there are two kinds of online follow relations for this user called two-way follow(TR) and one-way follow(OR)respectively. The two-way follow relationship means on one side is the user himself, on the other side is another micro-blog user and both are followers to each other. We investigated the data from 5000 authorized real sina Weibo users from API of *sina* micro-blogging service, from which we observed:1)Most of the users have a low ratio of two-way follow. 2)The two-way follow relation is basically between real life social circles such as friends, classmates, colleagues that have relatively close relation. Hence, we choose to use the two-way online follow relationship to generate shortcuts. More specifically, if two users in a group maintain a two-way online follow relationship, a virtual and regular directed line is established between nodes which these two users mapped into.

Assume there is a shortcut between user *i* and user *j* in the binary T-space. In shortcut based friend recommendation, we first find interest similarity users for user *j* by using match detection method in the cube. Then just recommend these users to user *i*. Note that the similarity levels of these users to user *i* are all added by one. In this way, more individuals have similar interest with user i are found.

V. SIMULATION

A. Data Collection

We collected data from *sina* Weibo, the most popular micro-blogging service in China. Weibo has published its APIs since2010 and these APIs allow us to get all the tweets from a user's different timelines. Since our algorithm is location based, we choose an ordinary urban area of *wuhou* district in *chengdu*, a city of China, for realistic data collection. We select sina Weibo users who work, study or live in the *wuhou* area as testing candidates. Moreover, we obtained authorization from 5000 real Weibo users including 1500 students, 2000 office workers and 2500resident in this area, who grant us full access to all the authentication-protection user data. We crawl their tweets history and get 56028 tweets written by these users from November21, 2013 to May 22, 2014 through Weibo's APIs.

We use the following metrics: precision(P), average precision at K(AP@K) , which are defined as:

$$P = \frac{N_{hit}}{m} \qquad (1)$$

$$AP@K = \frac{\sum_i^K P(i)}{N_{hit}} \qquad (2)$$

where m is the size of the recommendation list, N_{hit} is the number of users in the recommendation list belonging to the top m relevant matches. *P(i)* means the precision at cut-off *i*in the recommendation list. Due to location restriction, only a limited number of interest match users can be found and thus we set the length of recommendation list as 5 in our simulation.

We compare our method with the following algorithms:(1)RR, where users are randomly chosen from candidates to generate the recommendation list; influence-based Recommendation (INFR).(2)In INFR, we recommend candidates based on their social influence, which is measured by the number of users' followers; (3)Content-based Recommendation (CR). In CR, we find *m* most similar items with the user's latest tweet from other users that located in the same place with him based on content similarity and recommend users by combining the recommendation results from the similar items.

B. Algorithm Performance Evaluation

As shown in table 1, binary-R significantly improves the friend recommendation in all the metrics. We draw the following conclusions from these results. First, RR barely shows any effect and the worst performance, which makes it clear that randomly choose some users to recommend has little effect in improving social connection among users. Second, the poor performance of INFR is because influential users may be neither have similar profile with the user nor share any social friendship relations with the user. Moreover, recommendation notification messages maybe easily neglected by influential users due to many these messages are received by them per day. Third, CR shows the best performance in all of our comparison algorithms, owing to its incorporate tweets content-based features during the recommendation.

TABLE I. RESULT COMPARISON

	binary-R	**CR**	**INFR**	**RR**
Precision	58.2%	24.4%	3.43%	0.8%
AP@K	46.6%	18.3%	2.5%	0.5%

We also test the performance of our algorithms when only set the recommend list as 1,2,3,4 respectively and compare them with our best comparison algorithm CR. The experimental results are shown in Figure3.When reducing the number of recommended users, we can see that binary-R always has the best performance compared with CR algorithms based on precision. Especially, when the recommend list is only 1, binary-R shows a remarkable improvement on precision, which confirms our binary-R performs much better when only a few users are recommended.

ACKNOWLEDGEMENT

This work is supported by the national natural science foundation of China (grant nos .61103227,61272526) and the Tianjin normal university research foundation for talented person(no.5RL133).

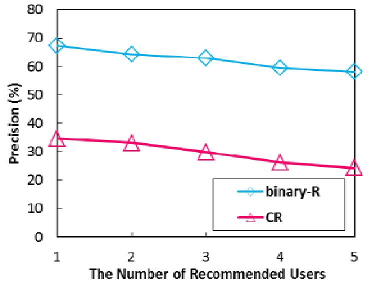

FIGURE III.RESULTS WITH CHANGE THE NUMBER OFRECOMMENDED USERS

REFERENCES

[1] Y.Chen, JC.Zhao, X.Hu, &XM.Zhang, From interest to function:location estimation in social media, *Proc. Int'27 AAAI Conference on Artifical Intelligence*, pp. 180–185,2013.

[2] W.Zhao, J.Jiang, J.Weng, H.Yan, &X.Li, Comparing twitter andtraditional media using topic models, *Advances in Information Retrieval,*pp. 338–349, 2011.

[3] X.Zuo, A.Chin, X. Fan, B.Xu, &DZ.Hong, Connecting people ata conference: a study of influence between offline and online using amobile social application, *Proc. Int GreenCom Conference on Green Computing and Communications*. pp. 277–284, 2012.

[4] L.Nowell, David, &J.Kleinberg, The linkprediction problem forsocial networks,*Journal of the American society for information science and techonology*.2011.

[5] DM.Blei, AY.Ng, & MI.Jordan, Latent dirichlet allocation,*Journal of machine learning research*. 2003.

[6] J. J, Z. F, & L. EP, Finding bursty topics from microblogs, *Proceedings of ACL*, pp. 536–544, 2012.

[7] M. Fire, Link prediction in social networks using computationallyefficient topological features. *Proc. Int Conference of Socialcom*, 2011.

[8] S. Zhang, H. Jiang, &J. Carroll, Integrating online and offlinecommunity through facebook,*Proc. Int CTS Conference on Collaboration Technologies and Systems,* 2011.