Zhu JQ, Lu L, Ma CM. From interest to location: Neighbor-based friend recommendation in social media. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(6): 1188–1200 Nov. 2015. DOI 10.1007/s11390-015-1593-3

From Interest to Location: Neighbor-Based Friend Recommendation in Social Media

Jin-Qi Zhu [1] (朱金奇), Li Lu [2] (鲁 力), *Member, CCF, ACM, IEEE*, and Chun-Mei Ma [2](马春梅)

[1] *School of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China*

[2] *School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731, China*

E-mail: jsjzhujinqi@mail.tjnu.edu.cn; luli2009@uestc.edu.cn; chunmeima2011@gmail.com

Received May 18, 2015; revised August 22, 2015.

Abstract Recent years have witnessed the tremendous development of social media, which attracts a vast number of Internet users. The tweets these users posted provide an effective way of understanding user behaviors. A large amount of previous work benefits from mining user interest to make friend recommendation. However, the potentially strong but inconspicuous relation between location and interest interaction among social media users is overlooked in these studies. Different from the previous researches, we propose a new concept named neighbor-based friend recommendation (NBFR) to improve the friend recommendation results. By recommending surrounding users who have similar interest to each other, social media users are provided a unique opportunity to interact with surrounding people they may want to know. Based on this concept, we first mine users' interest from short tweets, and then propose to model the user interest with multiple topics under the hypercube structure for friend recommendation. At the same time, we also offer a topic matching shortcut algorithm for more extensive recommendation. The evaluations using the data gathered from the real users demonstrate the advantage of NBFR compared with the traditional recommendation approaches.

Keywords micro-blogging system, neighbor, friend recommendation, hypercube

1 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]. A large number of content users posted present us a big data environment in social media platforms. In micro-blogging systems like Twitter and Weibo, users are provided with a powerful and convenient means of adding new friends, spreading breaking news, sharing information, performing virtual social activities and so on.

One of the most fundamental functions of micro-blogging systems is to enhance online social interaction among Internet users. To enhance users' virtual contacts, friend recommendation is considered as one of the most fundamental topics and has drawn much research efforts in recent years. For instance, in [3], a friend recommendation algorithm was designed and only those that have high attention were recommended to each other. In [4], the authors used link prediction based on "friend of a friend" approach to suggest one human being to another. The authors in [5] made a comparison between recommendations that are based on a user's familiarity network and his/her similarity network. The authors in [6] recommended Twitter users to follow using collaborative filtering approaches.

Regular Paper

Special Section on Networking and Distributed Computing for Big Data

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61103227, 61172185, 61272526, 61472068, and 61173171, the China Postdoctoral Science Foundation under Grant No. 2014M550466, the Introduced Research Funds for Tianjin Normal University under Grant No. 5RL133, and the Tianjin Research Program of Application Foundation and Advanced Technology under Grant No. 15JCQNJC01400.

These approaches, however, do not consider the relationship among the tweets generated by social media users, making them hard to find the truly useful information they are really interested in.

In fact, the high-dimensional content posted by millions of users presents both opportunities and challenges to the contemporary research. Regarding microblog users as social sensors, we can collect tremendously large datasets to facilitate the understanding of user behaviors. Part of previous efforts benefit from the knowledge of user interest to improve the friend recommendation result. For example, Zuo *et al.*[7] found people with similar characteristics are more likely to form ties with each other. The authors in [8] demonstrated that the contact among interest similar people occurs at a higher rate than that among dissimilar people. Chen *et al.*[9] studied people recommendations designed to help users find known contacts and discover new friends on social networking sites. Hsu *et al.*[10] addressed the problem of link recommendation in weblogs and similar social networks by proposing an approach based on collaborative recommendation using the link structure of a social network and content-based recommendation using mutual declared interests. It seems that interest-based recommendation realizes the need for boosting users' social contacts, while in fact recent study[7] shows that users with similar interest still contact with a low frequency.

Moreover, although many interest-based studies have been devoted in the previous work, the potentially strong but inconspicuous relation between users' interest and users' location is overlooked. In fact, a tweet often reflects the posting user's interest or behavior, and given users with similar interest and located in the same location, they are more plausible to establish interaction with each other. For example, user "Bob" likes traveling. He finds another user "Alice" who locates in the same place with him is also fond of traveling. Since people within the same place are highly likely to be offline encounters, if "Bob" further finds that "Alice" is over here and is just his type, he will then contact with "Alice" and keep in touch with her in the near future with a high probability. None of previous studies can be directly applied to the above mentioned task because they have not related surrounding users to their interest.

Inspired by these observations, we propose in this paper a new approach named neighbor-based friend recommendation (NBFR), which enables users with similar interest to interact with one another in the same physical domain. Since people within the same place are highly likely to be offline encounters, by interest similarity recommending, we provide users an effective way of contacting with people around them and whom they may be interested in. Nowadays, more and more people join the Internet and become the online users. Thus, our neighbor-based friend recommendation also helps bridge the gap between the real and the virtual. NBFR not only can be implemented as an app and run on smartphones but also can be added as a function of micro-blogging systems to enhance their usability in the future. There are three steps of achieving this goal. First, we generate the user topic interest matrix to mine users' interest from tweets contents information. Second, we try to measure users' interest similarity based on their interest vectors. Finally, matched users are ranked and recommended according to their similarity levels.

In order to measure user interest match, we propose to utilize the hypercube to explain the solution of users with multiple interest topics. We map the topics that we select for distinguishing a user into a hypercube space. Each topic is represented as a component of the vertex's coordinate in the cube. Considering that users are usually only interested in a small subset of topics, we prefer to construct a binary hypercube rather than a general cube because the former one could more accurately measure the interest similarity between two users. We also propose a topic matching shortcut algorithm for more extensive friend finding.

The main contributions of this paper include:

• We propose a new concept named neighbor-based friend recommendation (NBFR). NBFR enables users with similar interest to interact with one another in the same physical domain. By combining users' interest and users' location together, we construct a bridge between users' online and offline interaction.

• We present a shortcut-based friend recommendation algorithm under the hypercube structure.

• We evaluate our algorithms based on data gathered from the real users. Experimental results prove that our approach achieves high performance.

The remainder of this paper is structured as follows. Section 2 gives the related work. Section 3 presents an overview of our NBFR. In Section 4, we present the user interest detection method in detail. Section 5 depicts the interest match process. We evaluate the performance of our method and present the results in Section 6. Finally, Section 7 concludes the paper.

2 Related Work

2.1 Recommendation Approaches

In the last few years, a number of researchers have made many contributions to big data development[11-14]. As the big data booms, recommendation approaches which are used to recommend documents, users or items that users may be interested in have been a fertile area of research. The commonly used recommend approach is Collaborative Filtering (CF) recommending[3]. CF recommending mainly relies on the rating given by users to predict users' preferences. However, CF systems are easily influenced by the unfair ratings and what kind of users is appropriate to follow is still a difficult problem. In paper [15], to enhance a tweet's diffusion by finding the right persons to mention, a recommendation scheme which adopts user interest match was proposed. The authors of [16] studied people recommendations designed to help users find known, offline contacts and discover new friends on social networking sites. In practical application, Facebook launched a feature called "people you may know" to recommend friends by using link predictions based on a "friend of a friend" approach[17]. The feature of "people you may know" assumes people know each other with a high probability if they have common friends. Micro-blog also has the similar feature and it can recommend friends through geographical location information, users' indirect follow information, or common friends information. But link prediction approaches may cause some negative effects such as "richer get richer".

2.2 Hypercube

The applications of hypercubes have been initially studied in parallel and distributed computing. Most of recent hypercube-based researches focus on routing. In [18], the author utilized hypercube to design a team multicast routing protocol to address the scalability in mobile ad hoc networks. Huo *et al.*[19] defined a routing selection and maintenance rule based on a logical hypercube structure. In [20], the authors linked Bluetooth devices as a hypercube to construct a parallel computation and communication environment. Trying to leverage hypercube properties, the authors in [21] mapped social features into hypercube to design routing algorithm for HCNs (human contact networks). In this paper, we also try to use hypercube to model and analyze user interest match solution in geometric view.

3 Method Overview

In this section, we introduce the whole processing of NBFR which integrates both users' location and users' interest together. For a certain user, candidates who locate around and have similar interest with him/her are detected and recommended. To achieve this goal, our approach mainly comprises of three key components: neighbor definition, user interest detection, and interest match and recommendation. The workflow of NBFR is shown in Fig.1.

Neighbor Definition. Neighbor definition can be achieved by using two different methods. First, in the outdoor, the GPS device equipped in the smartphone is used to obtain the user's location which is then uploaded to a central server. Thus, users can be divided into different neighbor groups according to their current

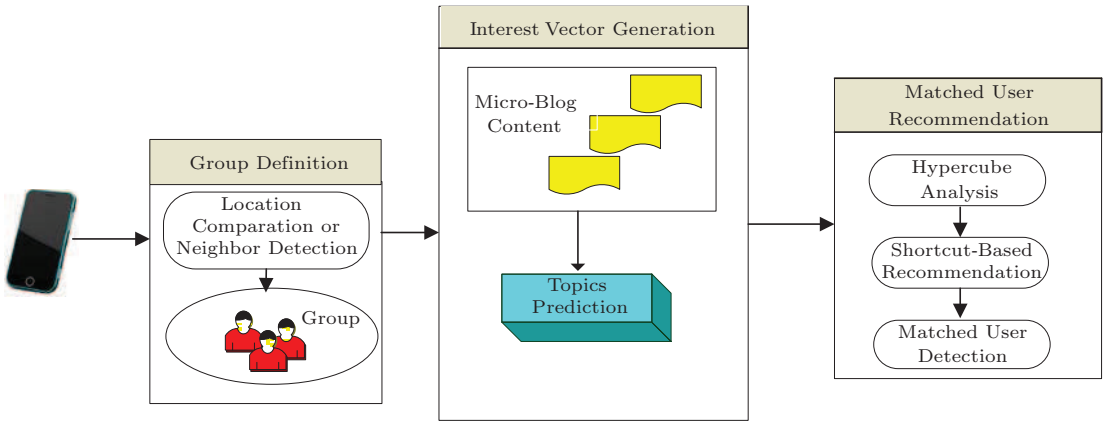


Fig.1. Framework of NBFR.

locations. Since the GPS value may be not accurate, we assume people within a certain relative distance are selected and classified as a same group. Moreover, the clearly visible distance of human eyes when being in barrier-free environment is about 300 meters[22]. Thus, we define the maximum relative distance in our group classification as 300 meters. Second, short-range wireless communication devices equipped in smartphone such as the Bluetooth or WiFi direct can be used for neighbor detection in indoor environment. Thus, we group users who can communicate directly via short-range communication devices into a neighbour group. Note that users in the same neighbor group see each other with high probability in the real world.

User Interest Detection. It is worth noting that users' tweets can reflect their personal interest. Based on this observation, we sort the posted tweets of each user in a group, and then employ matrix factorization approach to construct user interest matrix which reflects users' hidden interest topic distribution.

Interest Match and Recommendation. Given user interest vectors, we map users in a group into a structured high-dimensional hypercube, in which the interest closeness between two users is measured. Finally, a recommendation list is generated based on the interest match results. The top N relevant candidates in the ranked list are recommended to the source user.

4 User Interest Detection

The match of users' interest is an intuitively important feature. To calculate the match, the largest challenge is to discover the hidden user interest from users' tweets, which differs from traditional interest distribution discovery because contents in the tweets are short and contain a wide variety of topics and even noise. Topic model clustering algorithms like LDA[23] are used in previous studies[24-25] to predict the topic distribution from tweets contents. In the standard LDA, a document contains a mixture of topics, represented by a topic distribution, and each word has a hidden topic label. This is a reasonable assumption for long documents. However, for short microblog posts, a single post is most likely to be about a single topic[1]. Therefore, we prefer to employ both bag-of-words and matrix factorization approaches to indicate users' topic interest from raw tweets in this section.

4.1 User Feature Matrix Construction

For users in a same group, we crawl their tweets and define d as the set of recent tweets they posted. For d, after word segment by *jieba* tool (a toolkit used for Chinese word split), we eliminate all the stop words in the posted tweets and only keep a word if it is identified as a noun. Thus, based on all these kept words, a dictionary with distinct words is constructed. Using the indexes of the dictionary, each tweet is represented by an n-entry vector, where n is the total number of words in the dictionary. Moreover, each entry of the vectors refers to the count of the corresponding entry in the dictionary. For example, in vector (1, 3, 1, 1, 2, 0, 0, 1, 0), the first entry is 1 because the first word in the dictionary appears in the corresponding tweet once. In this way, each user in the group can be denoted as a matrix $\boldsymbol{X}_{m\times n}$ as shown in (1), where each row represents a tweet vector, m is the number of tweets for the user and the i-th word in the dictionary appears in tweet j with a frequency of $f_{ij}(i = 1, 2, \ldots, n)$.

$$\boldsymbol{X} = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1j} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2j} & \cdots & f_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mj} & \cdots & f_{mn} \end{pmatrix}. \tag{1}$$

For user a, if we sum each column of its matrix $\boldsymbol{X}_a$, then a new row vector $\boldsymbol{y}_a = (x_{a1}, x_{a2}, \ldots, x_{ai}, \ldots, x_{an})$ is got, in which x_{ai} is the total number that the i-th word in the dictionary appears in user a's crawled tweets. In this way, all the users we crawled could be mapped into a matrix $\boldsymbol{Y}_{t\times n}$ as shown in (2).

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_t \end{pmatrix}, \tag{2}$$

where t denotes the total number of users crawled in the social media and $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, $\ldots$, $\boldsymbol{y}_n$ denote the new row vectors we got for users (like $\boldsymbol{y}_a$ for user a).

4.2 Matrix Factorization for User Interest Detection

Matrix factorization technique is a popular method in discovering the latent features underlying the interactions between users and items. One obvious application is to predict ratings in collaborative filtering. The intuition behind matrix factorization for rating prediction is that there should be some latent features which

determine how a user rates an item. If we can discover these latent features, we should be able to predict a rating with respect to a certain user and a certain item, because the features associated with the user should match with the features associated with the item. Since matrix factorization works by users' feature discovery, here we try to use matrix factorization to indicate user interest distribution from social media.

Consider we want to extract k hidden user interest topics from matrix $\boldsymbol{Y}_{t\times n}$, where value k satisfies $k < t$ and $k < n$. Then we need to find two matrices $\boldsymbol{U}_{t\times k}$ and $\boldsymbol{V}_{n\times k}$ so that their product approximates matrix $\boldsymbol{Y}_{t\times n}$. Each row of $\boldsymbol{U}_{t\times k}$ will be the strength of the associations between a user and the interest topics.

$$\boldsymbol{Y}_{t\times n} \approx \boldsymbol{U}_{t\times k} \cdot \boldsymbol{V}_{n\times k}^{\mathrm{T}}.$$

Suppose the prediction of a rating of an item V_j by user U_i is denoted as the following equation:

$$Y'_{ij} = \boldsymbol{U}_i^{\mathrm{T}} \boldsymbol{V}_j,$$

then, to get the matrix $\boldsymbol{U}_{t\times k}$, the difference between the estimated rating Y'_{ij} and the real rating Y_{ij} should be minimized as:

$$\min(Y_{ij} - \boldsymbol{U}_i^{\mathrm{T}} \boldsymbol{V}_j)^2. \tag{3}$$

Suppose the difference between the estimated rating Y'_{ij} and the real rating Y_{ij} follows the normal distribution. The objective function of (3) can be deduced and modeled as:

$$L = \sum_{i=1}^{n} \sum_{j=1}^{t} (Y_{ij} - \boldsymbol{U}_i^{\mathrm{T}} \boldsymbol{V}_j)^2. \tag{4}$$

Furthermore, to avoid overfitting caused by the sparse character of matrix $\boldsymbol{Y}$, regularization is introduced in our algorithm. This is done by adding a parameter λ which denotes the weight of regularization terms. Thus, (4) is modified as (5):

$$L = \sum_{i=1}^{n} \sum_{j=1}^{t} (Y_{ij} - \boldsymbol{U}_i^{\mathrm{T}} \boldsymbol{V}_j)^2 + \lambda(\|\boldsymbol{U}_i\|^2 + \|\boldsymbol{V}_j\|^2). \tag{5}$$

Our objective is then to minimize function L. To achieve this, stochastic gradient descent is used and the update rule of $\boldsymbol{U}_i$ is given by:

$$\boldsymbol{U}_i = \boldsymbol{U}_i - \beta \frac{\partial L}{\partial \boldsymbol{U}_i}, \tag{6}$$

where β is a constant whose value determines the rate of approaching the minimum. Note that

$$\frac{\partial L}{\partial \boldsymbol{U}_i} = 2\lambda - 2 \sum_{i=1}^{n} \sum_{j=1}^{t} \boldsymbol{V}_j (Y_{ij} - \boldsymbol{U}_i^{\mathrm{T}} \boldsymbol{V}_j). \tag{7}$$

Using (7) to rewrite (6), we acquire (8) as:

$$\boldsymbol{U}_i = \boldsymbol{U}_i - 2\beta \left(\lambda - \sum_{i=1}^{n} \sum_{j=1}^{t} \boldsymbol{V}_j (Y_{ij} - \boldsymbol{U}_i^{\mathrm{T}} \boldsymbol{V}_j) \right). \tag{8}$$

Suppose we set $k = 10$, which is an empirical value of interest categories. The stochastic gradient descent process is stopped when $L < 0.001$ in the experiment. After the stochastic gradient descent process, we get the matrix $\boldsymbol{U}_{t\times k}$ which indicates users' topic distribution. Each row of matrix $\boldsymbol{U}_{t\times k}$ is a vector that describes a user's topic interest in the group. Suppose user u_i's topic interest we get from $\boldsymbol{U}_{t\times k}$ is $\boldsymbol{u}_i = (u_{i1}, u_{i2}, \ldots, u_{ik})$, $i = 1, 2, 3, \ldots, t$,

$$w_{ij} = \frac{u_{ij}}{\sum_{j=1}^{k} u_{ij}}, \quad j = 1, 2, \ldots, k,$$
$$w_{i1} + w_{i2} + w_{i3} + \ldots + w_{ik} = 1.$$

5 User Interest Match

In this section, we explain user interest similarity with hypercube. In order to achieve our goal, we map the multiple topics of users we get from Section 4 into topic hypercube, a structured topic space.

5.1 Topic Hypercube Model

5.1.1 Multidimensional Hypercube Construction

Generally, as we all know, with one interest topic only, we can distinguish the interest of two different users in one-dimensional (1D) space. For example, as shown in Fig.2, the difference between users A and B is Δu. If there are two topics provided, the difference between the two users could be presented by a two-dimensional (2D) vector (Δu, Δv). Similarly, if there are three or more than three interest topics, we have to describe the difference of various users by using multipledimensional space. Each topic of the user is represented as a component of the vertex's coordinate in the hyper space.

Fig.3 represents a $3\times2\times2$ hypercube which consists of 12 users. In this example, there are three different interest topics in this cube, named sports, health, and traveling respectively. Originally, user A and user B have their own interest topic vectors as A:($x1$, $x2$, $x3$) and B:($y1$, $y2$, $y3$). With the aim of mapping users into the hypercube, we set user A as the coordinate origin (0, 0, 0) of the hypercube. By computing the relative distance between vectors ($x1$, $x2$, $x3$) and ($y1$, $y2$, $y3$), user B is mapped into vertex (2, 1, 1) in the cube. In

this way, we plot users with their corresponding vertexes in the cube.

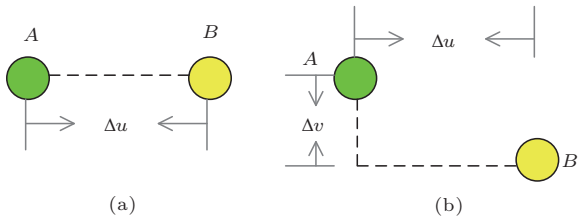


Fig.2. Distinguishing users with one topic and two topics provided. (a) 1D. (b) 2D.

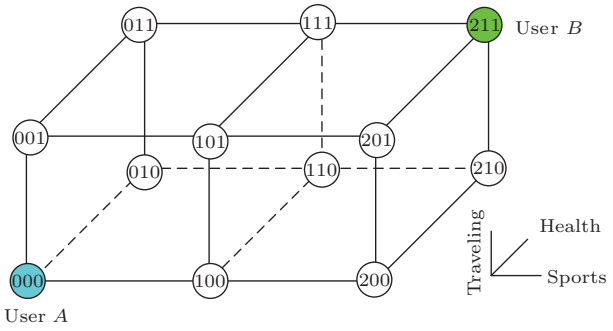


Fig.3. 3D hypercube.

Moreover, assume that there are N users in a group, each user is represented by a vector $(T_1, T_2, \ldots, T_{10})$, a representation of his/her interest distribution. According to the above analysis, these users could be mapped into a structured 10-dimensional hypercube (or simply 10D cube), in which each node (vertex) represents a user in the group. Thus, there is usually a one-to-one correspondence between a user and a node in the cube. In the 10D cube, two nodes are connected if and only if they differ in one topic.

To map a group of nearby users into the 10D cube, we should acquire the relative distance between two user vectors. Suppose there are two users A and B, with their vectors as $(x_{a1}, y_{a2}, z_{a3}, \ldots, w_{a10})$ and $(x_{b1}, y_{b2}, z_{b3}, \ldots, w_{b10})$ respectively. We can calculate their relative distances for each component as:

$$\begin{cases} \Delta x_{ab} & = \|x_{a1} - x_{b1}\|, \\ \Delta y_{ab} & = \|y_{a1} - y_{b1}\|, \\ \Delta z_{ab} & = \|z_{a1} - z_{b1}\|, \\ & \vdots \\ \Delta w_{ab} & = \|w_{a1} - w_{b1}\|. \end{cases} \tag{9}$$

With the obtained relative distance vector $(\Delta x_{ab}, \Delta y_{ab}, \Delta z_{ab}, \ldots, \Delta w_{ab})$, we can map A and B into a topic space (T-space). Obviously, the 10 components are totally different kinds of topics. For example, x is entertainment, y is sports and z denotes health.

Fig.4 shows an example of constructing the T-space based on three diverse interest topics. Suppose user A has a 3D interest vector (0.12, 0, 0.1) and B has an interest vector as (0.4, 0.24, 0). According to (9), the relative distance vector between A and B is (0.28, 0.24, 0.1). If user A is set as the coordinate origin (0, 0, 0), in order to map user B into the T-space, we have to

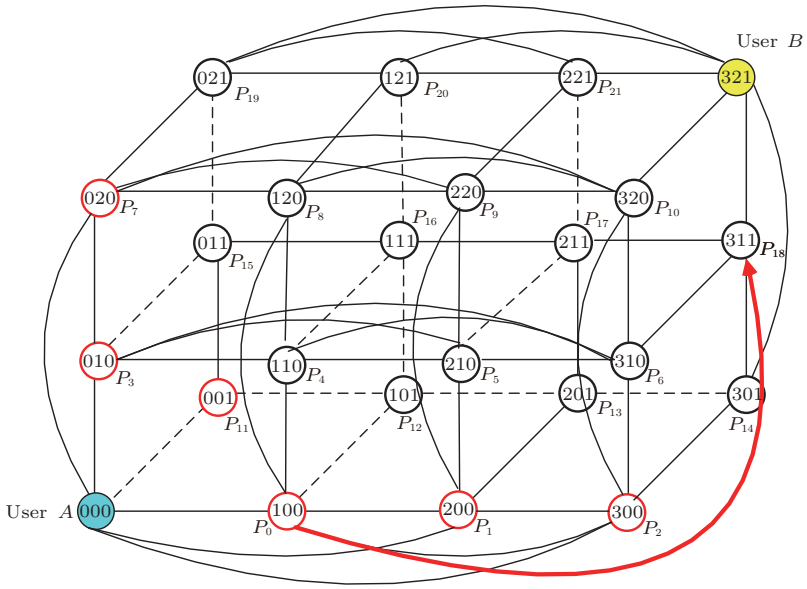


Fig.4. Example of the T-space.

deal with each component of the relative distance vector as follows. First, we round each component value up to one decimal. Then, each component value is enlarged by multiplying 10. After these operations, user B is mapped into vertex (3, 2, 1) in the cube, as shown in Fig.4. Meanwhile, we note that two users have a connection if they differ in exactly one topic.

5.1.2 Match Detection in T-Space

The topic distance between node i and node j in the T-space is denoted as D_{ij}, which is the hamming distance between users i and j. For instance, the topic distance from node A to node P_4 is 2 in Fig.4. Generally, for two nodes in the cube, the more interest topics in common, the less their topic distance. Hence, we use topic distance as the metric to measure the interest match between two micro-blog users in a group.

Taking user A in Fig.4 for example, by comparing the topic distance from each user to user A in the T-space, we rank the others in the same group with user A into different similarity sets and users in the same set have the same similarity level. More specifically, we detect from the cube that:

• Users who are most interest relevant with user A are ranked to set $C^0 = (P_0, P_1, P_2, P_3, P_7, P_{11})$, in which $D_{AP_i} = 1$, for $i = 0, 1, 2, 3, 7$ and 11.

• For each node in set $C^1 = (P_4, P_5, P_6, P_8, P_9, P_{10}, P_{12}, P_{13}, P_{14}, P_{15}, P_{19})$, its topic distance to user A is 2.

• Users who have no similar interest with A belong to set C^2 and $C^2 = (P_{16}, P_{17}, P_{18}, P_{20}, P_{21}, B)$.

In summary, we convert the user interest match problem in human social space into a multidimensional hypercube space. By taking advantage of the structural property of the hypercube, it is efficient to discover interest closeness among users.

5.2 Binary Hypercube Model

5.2.1 Binary Hypercube Construction

In Subsection 5.1, we discussed user interest match in the general T-space. However, there are two main problems when performing user interest match in the constructed T-space. First, generally, since user interest vectors have different dimensional values, it is not very easy to measure interest similarity for users. Second, the relative distance cannot represent the distinct topic difference between two users. To enhance matching performance, we propose a binary cube structure in this subsection.

The binary hypercube is a special cube in which each topic has a binary value: 0 or 1. To construct a binary cube, we should change topic 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 topic interest vectors into binary vectors, two steps are taken as follows. Firstly, key topics are selected from each interest vector. Secondly, we change the component value of each key topic to 1, and let the values of the other topics be 0. The first step is done by transforming component values of each user vector less than threshold γ to 0 and keeping the component values if and only if they are not smaller than γ. Note that the components of the users' topic vector have two features: 1) most of the components have values less than 0.1; 2) only a few component values are larger than 0.1. Thus, we let the threshold γ be 0.1 in our simulation. Accordingly, given $\boldsymbol{W}$ = (0, 0.21, 0.06, 0.30, 0.05, 0.11, 0.03, 0.02, 0.09, 0.13), $\boldsymbol{W}$ is transformed to binary vector $\boldsymbol{W}^*$ = (0, 1, 0, 1, 0, 1, 0, 0, 0, 1) by taking these steps.

After mapping binary vectors into the hypercube, a 10D binary cube is formed (binary T-space). More specifically, at each dimension in the binary cube, users are separated based on whether they have identical key interest topics or not. Moreover, users with the same key topics are mapped into the same vertex in the cube. In this way, a general T-space is "compressed" into a binary T-space, even though each topic may have many different values. Unlike in general T-space, there is no one-to-one correspondence between a micro-blog user and a node in the binary cube. Fig.5 shows a concrete 4D binary cube constructed based on four different interest topics. In Fig.5, 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. If the other users have a different topic value than node 0000 in the dimension, the corresponding bit is set to 1. Note that in the binary cube, two nodes are directly connected if they differ in only one key topic.

5.2.2 Match Detection in Binary Cube

Match detection in the binary cube can be achieved as follows. 1) We select the coordinate of the vertex which the source user is mapped into. 2) We select the coordinate of a vertex from the cube at a time. This vertex must have users mapped into it. This is done

till all the vertexes in the cube are chosen. 3) We compute the bitwise AND operation of the two coordinate values selected in 1) and 2). At last, we count the total number of value 1 in the result. The more values of 1 in the final result, the higher the similarity level of the users that the associated vertex represents.

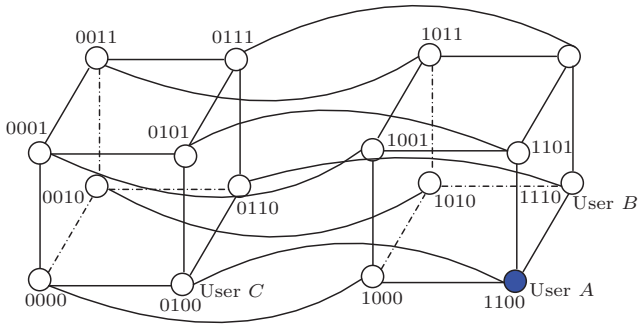


Fig.5. Binary hypercube with four topics.

For example, user B is mapped into node 1110 and user C is mapped into node 0100 in Fig.5 respectively. To recommend friends for user A (1100), we compute the bitwise AND operation of values 1100 and 1110, and get result 1100. Thus, user B is set to level 2 since the number of value 1 in 1100 is 2. Similarly, user C is set to similarity level 1 as a result. Thus, user B is more interest relevant with user A than user C.

In the binary cube, since users with the same key topics are mapped into the same node and users are ranked based on the number of identical key interest topics, user interest match in the structured binary T-space is more efficient and accurate compared with that in the general T-space.

5.3 Shortcut-Based Recommendation

We now consider a new issue of our interest 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/she only logs in his/her micro-blog and posts a tweet once a month. Thus, user topic interest vectors we generate in Section 4 may not precisely reflect the profile of users. In [26], the authors depicted that a person's interest can impact his/her friends' interest in the physical world, and vice versa. Zhang *et al.*[27] proposed that offline friends usually have potential hobbies in common. Therefore, in terms of a more precise interest recommendation, we propose to use offline friendship for more extensive user interest discovery.

Here, we propose the idea of shortcut. Traditionally, in the general T-space and the 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, a controlled jump to a micro-blog user that is more than one topic distance away in the binary cube is allowed. Such a controlled jump is called a shortcut. As illustrated in Fig.4, if users P_0 and P_{18} are offline friends, the path from P_0 to P_{18} is then generated as a shortcut. From Fig.6, we can consider a virtual directed line with two nodes: A and B. A and B differ in dimensions i, $i+1$, ..., j. When $j = i+1$, it corresponds to a regular directed line with A and B differing in exactly one topic.

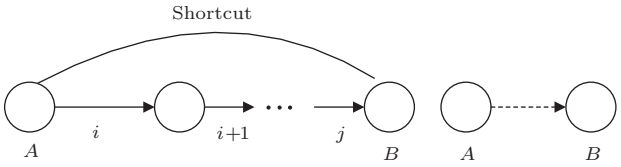


Fig.6. Illustration of shortcut.

Intuitively, given the online friend information of a micro-blog 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 that on one side is the user himself, on the other side is another micro-blog user, and both are followers to each other. Table 1 investigates the data from 5 000 authorized real Sina Weibo users from API of Sina micro-blogging service, from which we observe: 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, and 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 the nodes which these two users are mapped into.

Assume there is a shortcut between user i and user j in the binary T-space. In shortcut-based friend recommendation, to recommend interest match users for user i, we first find interest match users for user j by using match detection method we adopted in the binary

cube. Then we just recommend these users to user i. In this way, more individuals that have similar interest with user i are found.

Table 1. Ratio of Users Having Two-Way Relation

	Ratio=TR/(TR+OR)	Number of Users in Each Type	Total Number of Users
1	0 ∼ 29%	Student: 1 024	3 159
		Worker: 1 298	
		Resident: 837	
2	30% ∼ 59%	Student: 339	986
		Worker: 246	
		Resident: 401	
3	60% ∼ 100%	Student: 137	855
		Worker: 456	
		Resident: 262	

6 Simulation

6.1 Data Collection

We collected data from Sina Weibo, the most popular micro-blogging service in China. Sina Weibo has published its APIs since 2010 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. The extracted real map of Wuhou district which covers 75.36 square kilometres and has about 1 260 000 people in total is shown in Fig.7. Many previous researches[28-29] validated that people have regular mobility, and two to four main locations cover more than 70% of the overall daily trips of a person. Thus, we select Sina Weibo users who work, study or live in the Wuhou area as testing candidates. Moreover, we obtained authorization from 5 000 real Sina Weibo users including 1 500 students, 2 000 office workers, and 1 500 residents in this area, who grant us full access to all the authentication-protection user data, including users' follows, followers, tweets and locations. We crawl their tweets history and get 56 028 tweets written by these users from November 21, 2013 to May 22, 2014 through Sina Weibo's APIs.

It is difficult to find the current GPS data from users' Weibo pages. However, Sina Weibo supports the per-tweet geo-tagged function at present. Generally, each posted tweet now is labeled with a geo-tag which records the street, the district and the city where the user stayed when posting the tweet. After sequencing each user's tweets in a time line order, we treat the latest tweet's geo-tag as his/her current location for simplicity. If two users have the same latest geo-tags, we assume they are within the same place. Since the geo-tags have fixed errors, the following test is done to investigate the errors of the geo-tags. We walked along several urban streets and posted a tweet every 200 metres. Table 2 records the testing results, from which we discovered the average error between the geo-tag and the owner's actual location reaches about 122.14 metres.

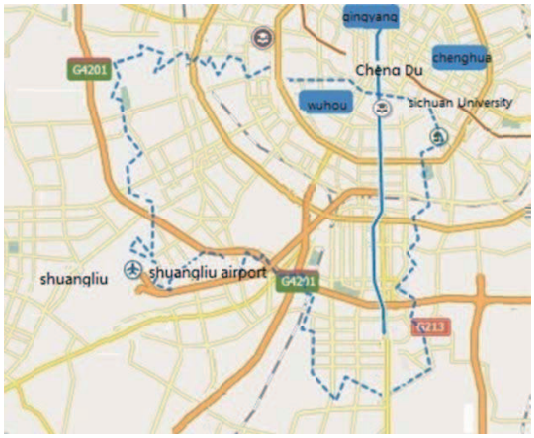


Fig.7. Simulation area.

Table 2. Geo-Tags Collection

Time	Street	Geo-Tag	Deviation (m)
1	Stadium Road	Stadium Road	–
2	Stadium Road	Stadium Road	50
3	Stadium Road	Stadium Road	50
4	Jingu Street	Stadium Road	10
5	Jingu Street	Jingu Street	–
6	Jingu Street	Dajian Road	200
7	Jingu Street	Jingu Street	–
8	Yarun Road	Jingu Street	200
9	Yarun Road	Jingu Street	400
10	Yarun Road	Jingu Street	600
11	Zijiang Road	Nanhuan Road	200
12	Zijiang Road	Zijiang Road	–

6.2 Evaluation Metrics

We use the following metrics: precision (P), average precision at K (AP@K)[15] and average hit rate (AHR), which are defined as:

$$P = \frac{N_{\text{hit}}}{m},$$

$$AP@K = \frac{\sum_{i=1}^{K} P(i)}{N_{\text{hit}}},$$

$$AHR = \frac{\sum_{u=1}^{U} \frac{N_{\text{talk}}}{N_{\text{hit}}}}{U},$$

where m is the size of the recommendation list, N_{hit} is the number of users who have similar interest with the source user in the recommend list. $P(i)$ means the precision at cut-off i in the recommendation list. N_{talk} is the number of users who belong to N_{hit} and successfully established connections with the source user. U is the total number of source users in our simulation. Due to location restriction, only a limited number of interest match users can be found and thus we set m as 5 in our simulation.

6.3 Comparison Algorithms

We compare Euclidean cube based recommendation (Euclidean-R) and binary cube based recommendation (binary-R) with the following algorithms.

• *Random Recommendation* (*RR*). In RR, users are randomly chosen from candidates to generate the recommendation list.

• *Influence-Based Recommendation* (*INFR*). In INFR, we recommend candidates based on their social influence, which is measured by the number of users' followers. We try to recommend the most influential users to a certain user.

• *Content-Based Recommendation* (*CR*). In CR, we find m most similar items with a user's latest tweet from other users that locate in the same place with him/her based on content similarity and recommend users by combining the recommendation results from the similar items.

6.4 Algorithm Performance Evaluation

As shown in Table 3, binary-R and Euclidean-R significantly improve the friend recommendation in all the metrics. We draw the following conclusions from these results.

Table 3. Result Comparison

	Binary-R (%)	Euclidean-R (%)	CR (%)	INFR (%)	RR (%)
P	58.2	37.50	24.4	3.43	0.81
AP@5	47.6	30.45	18.3	2.50	0.52
AHR	19.8	12.50	8.60	1.80	0.13

Note: P stands for precision.

First, RR shows the worst performance, which denotes randomly choosing some users to recommend has little effect on improving social connection among users. Second, the poor performance of INFR is because influential users may neither have similar profile with the user nor share any social friendship relations with the user. Moreover, recommendation notification messages may be easily neglected by influential users because they may receive a large number of recommendation messages per day. Third, CR shows the best performance in all of our comparison algorithms, owing to it considering tweet content.

Finally, our binary-R outperforms all the other algorithms. Compared with CR, it shows 34% increase in precision, 26% increase in AP@5 and 11.2% increase in AHR. This benefits from a careful design of user interest model, a user interest match function based on the feature of hypercube and the use of shortcut in potential friend finding. Moreover, binary-R has better performance than Euclidean-R. This is mainly because the Euclidean T-space is a sparse cube. On the contrast, since users with the same key topics are mapped into the same node in the binary cube, it is much easier to find interest similar users. Moreover, the AHRs in all the algorithms are low. This is reasonable and caused by some actual reasons. For instance, some users are not likely to talk with unknown people and thus friend recommendation notifications are treated as advertising messages and ignored by those users or some people may temporarily close friend recommendation notifications due to security reasons. However, due to the high friend recommendation precision of our algorithm, once a user opens a notification message, he/she shows a high probability of contacting with recommended users. This further confirms the effectiveness of our algorithms on boosting the social interaction among neighboring micro-blog users.

6.5 Limited Recommended Users

We also test the performance of our algorithms when setting the size of the recommend list as 1, 2, 3, 4 respectively. The experimental results are shown in Figs.8(a) and 8(b).

When reducing the number of recommend users, we can see that binary-R always has the best performance compared with both Euclidean-R and CR algorithms based on all metrics. Especially, when the size of the recommend list is only 1, binary-R shows a remarkable improvement on both precision and average hit ratio (AHR), which confirms that our binary-R performs much better when only a few users are rec-

ommended. Since the Euclidean cube is a sparse T-space, Euclidean-R does not perform so well as binary-R. However, Euclidean-R has better performance than CR. The AHRs of all algorithms drop when recommending fewer users, showing that establishing social ties is a quite difficult task. Recommending only few users will incur higher miss rate of notification messages, which leads the probability of establishing social connections among unknown users (AHR) to decrease.

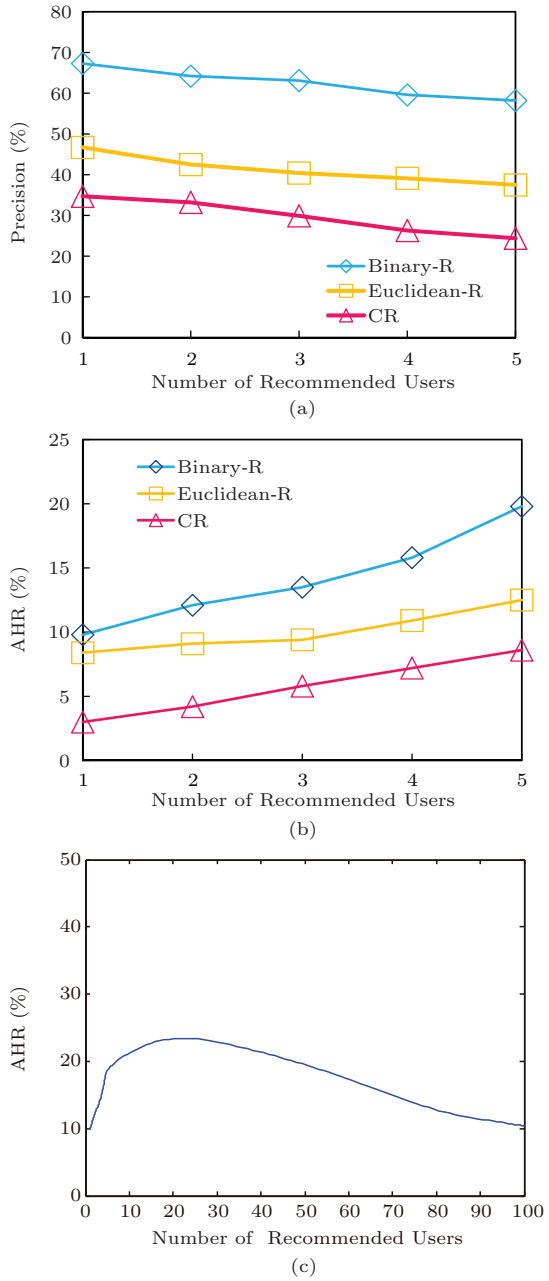


Fig.8. Results with changing the number of recommended users. (a) Precision. (b) AHR. (c) AHR vs the number of recommended users.

We also note the AHR value decreases as the recommended users increase in Fig.8(c). This is mainly due to that the precision of recommendation decreases when the number of recommended users increases. The unprecise recommendation may be treated as an interruption and neglected by the user.

6.6 Feature Importance Evaluation

To learn how features used in our system contribute to the user interest vector, we design this experiment to learn how shortcut-based recommendation contributes to the performance of our binary-R algorithm. The results are listed in Table 4.

Table 4. Effect of Shortcut-Based Recommendation on the Performance of Binary-R

	All (%)	No Friendship (%)
Precision	58.2	56.92
AP@5	47.6	45.15
AHR	19.8	19.21

When eliminating friendship (shortcut), the three metrics all decrease. Since there may exist no physical friend for a certain user or the number of physical friends for the user is small in the group he/she stays in, the three metrics only decrease slightly.

6.7 Relative Distance Evaluation

To the best of our knowledge, no previous studies have been done on the neighbor-based friend recommendation task. Thus, for simplicity, we assume the latest tweet's geo-tag of each user as his/her current location. To further evaluate the effect of distance parameter on the performance of recommendation, we select users in the dataset who work in Baoli center building as source users and assume they have the same GPS values. Moreover, remaining users in the dataset that meet the following conditions are chosen as recommended candidates. First, they work, study or live in the same street with the source users. Second, their working, studying or living place reflected in the map is within 300 meters from the location of the source users. The experimental results with different distance parameters are shown in Fig.9.

With the increment of the relative distance, we can see both the precision and AP@5 increase. This is because more candidates take part in friend recommendation. However, the AHRs change slightly as varying

the relative distance. This is mainly because people within a short relative distance know each other with a high probability in the physical world. Thus, they like to contact with each other. However, when increasing the relative distance value, though more users are recommended to the source users, the notification messages which recommend strangers may be ignored by the source user. This will lead to a limited AHR value when the relative distance is long.

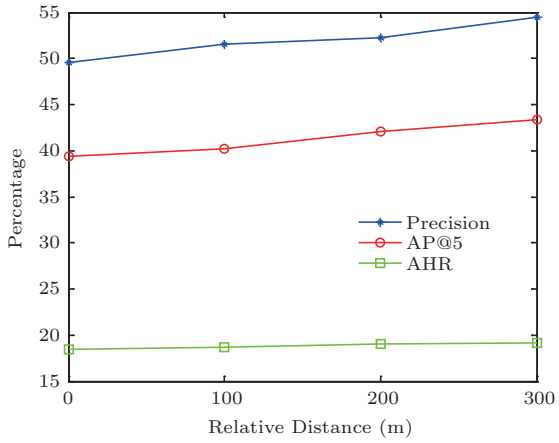


Fig.9. Relative distance to the recommend performance.

In short, all the experimental results may seem a bit low, which is in accordance with our expectation. The main reason is that both precise recommendation match among a certain number of people in real world and attracting strangers to establish connections are not easy jobs. However, by comparing our algorithms with some traditional friend recommendation algorithms, our binary-R algorithm performs well with the remarkable improvement on all metrics.

7 Conclusions

In this paper, by combining users' interest and users' location, we proposed a neighbor-based friend recommendation method, NBFR, which enables interest similar users to interact with one another in a dynamic, ephemeral environment. To achieve this goal, we deeply looked into tweets content to mine users' interest on one hand. On the other hand, we mapped the interest topics that we selected in distinguishing users into the hypercube space. Matched users are detected and ranked based on the inherent characteristic of the hypercube structure. NBFR can be added as a new function of micro-blogging systems to enhance their usability in the future. Many future studies can be further explored. For example, how to use short-range wireless communication devices for precise neighbor definition needs to be further discussed. It is also very necessary to study the security problem caused by using NBFR.

References

[1] Zhao W J, Jiang J, Weng J *et al.* Comparing Twitter and traditional media using topic models. In *Proc. the 33rd ECIR*, April 2011, pp.338-349.

[2] Chen Y, Zhao J C, Hu X *et al.* From interest to function: Location estimation in social media. In *Proc. the 27th AAAI Conference on Artificial Intelligence*, July 2013, pp.180-186.

[3] Moricz M, Dosbayev Y, Berlyant M. PYMK: Friend recommendation at MySpace. In *Proc. the 2010 ACM SIGMOD Int. Conf. Management of Data*, June 2010, pp.999-1002.

[4] Kazienko P, Musial K, Kajdanowicz T. Multidimensional social network in the social recommender system. *IEEE Trans. System, Man and Cybernetics, Part A: Systems and Humans*, 2011, 41(4): 746-759.

[5] Deng Z W, He B W, Yu C C, Chen Y X. Personalized friend recommendation in social network based on clustering method. In *Proc. the 6th ISICA*, Oct. 2012, pp.84-91.

[6] Hannon J, Bennett M, Smyth B. Recommending Twitter users to follow using content and collaborative filtering approaches. In *Proc. the 4th ACM Int. Conf. Recommender Systems*, September 2010, pp.199-206.

[7] Zuo X, Chin A, Fan X G *et al.* Connecting people at a conference: A study of influence between offline and online using a mobile social application. In *Proc. GreenCom/iThings/CPSCOM*, Nov. 2012, pp.277-284.

[8] Mcpherson M, Smith-Lovin L, Cook J. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001, 27: 415-444.

[9] Chen J, Geyer W, Dugan C, Muller M. Make new friends, but keep the old: Recommending people on social networking sites. In *Proc. the 27th Int. Conf. Human Factors in Computing Systems*, April 2009, pp.201-210.

[10] Hsu W H, King A L, Paradesi M S R *et al.* Collaborative and structural recommendation of friends using weblog-based social network analysis. In *Proc. AAAI Conference on Computational Approaches to Analyzing Weblogs*, March 2006, pp.55-60.

[11] Wen Y G, Zhu X Q, Rodrigues J P C, Chen C W. Cloud mobile media: Reflections and outlook. *IEEE Trans. Multimedia* (*TMM*), 2014, 16(4): 885-902.

[12] Hu H, Wen Y G, Chua T S, Li X L. Towards scalable systems for big data analytics: A technology tutorial. *IEEE Access Journal*, 2014, 2: 652-687.

[13] Xia W F, Wen Y G, Foh C H *et al.* A survey on software-defined networking. *IEEE Commun. Surveys and Tutorials*, 2015, 17(1): 27-51.

[14] Hu H, Wen Y G, Chua T S *et al.* Community-based effective social video contents placement in cloud-centric CDN network. In *Proc. the IEEE Int. Conf. Multimedia and Expo* (*ICME*), July 2014.

[15] Wang B D, Wang C, Bu J *et al*. Whom to mention: Expend the diffusion of tweets by @ recommendation on micro-blogging systems. In *Proc. the 22nd ACM Int. Conf. WWW*, May 2013, pp.1331-1340.

[16] Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 2007, 58(7): 1019-1031.

[17] Zhang L Z, Fang H, Ng W K, Zhang J. IntRank: Interaction ranking-based trustworthy friend recommendation. In *Proc. the 10th Int. Conf. IEEE TrustCom*, Nov. 2011, pp.266-273.

[18] Wu J. Distributed System Design (1st edition). CRC Press, 1998.

[19] Huo H W, Shen W, Xu Y Z, Zhang H K. Virtual hypercube routing in wireless sensor networks for health care systems. In *Proc. the 1st IEEE ICFIN*, Oct. 2009, pp.178-183.

[20] Chang C Y, Chang C Y, Sheu J P. BlueCube: Constructing a hypercube parallel computing and communication environment over Bluetooth radio systems. *Journal of Parallel and Distributed Computing*, 2006, 66(10): 1243-1258.

[21] Wu J, Wang Y. Hypercube-based multi-path social feature routing in human contact networks. *IEEE Trans. Computers*, 2014, 63(2): 383-396.

[22] Fuller H Q, Fuller R M, Fuller R G. Physics, Including Human Applications (1st edition). Longman Higher Education Press, 1978.

[23] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 2003, 3: 993-1022.

[24] Wang X Y, Sun L F, Wang Z, Meng D. Group recommendation using external followee for social TV. In *Proc. the 2012 IEEE ICME*, July 2012, pp.37-42.

[25] Chen J T, She J. An analysis of verifications in microblogging social networks — Sina Weibo. In *Proc. the 32nd ICDCSW*, June 2012, pp.147-154.

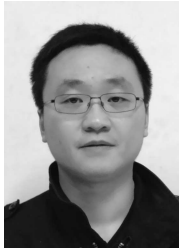
[26] Fire M, Tenenboim L, Lesser O *et al*. Link prediction in social networks using computationally efficient topological features. In *Proc. the 3rd PASSAT*, Oct. 2011, pp.73-80.

[27] Zhang S K, Jiang H, Carroll J M. Integrating online and offline community through Facebook. In *Proc. the 2011 IEEE Int. Conf. Collaboration Technologies and Systems*, May 2011, pp.569-578.

[28] Hsu W J, Spyropoulos T, Psounis K, Helmy A. Modeling time-variant user mobility in wireless mobile networks. In *Proc. the 26th IEEE INFOCOM*, May 2007, pp.758-766.

[29] González M, Hidalgo C, Barabási A. Understanding individual human mobility patterns. *Nature*, 2008, 453(7196): 779-782.

Jin-Qi Zhu received her Ph.D. degree in computer science from University of Electronic Science and Technology of China, Chengdu, in 2009. She is currently an associate professor in the School of Computer and Information Engineering at Tianjin Normal University, Tianjin. Her research interests include parallel and distributed computing, mobile and wireless computing, wireless sensor networks, and vehicular ad hoc networks.



Li Lu received his Ph.D. degree in computer science from the State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing, in 2007. He is currently an associate professor in the School of Computer Science and Engineering in the University of Electronic Science and Technology of China, Chengdu. His research interests include parallel and distributed computing, RFID technology, and wireless network security. He is now a member of CCF, ACM, and IEEE.

Chun-Mei Ma is now a Ph.D. student in the School of Computer Science and Engineering in the University of Electronic Science and Technology of China, Chengdu. Her research interests include parallel and distributed computing, mobile and wireless computing, wireless sensor networks, and vehicular ad hoc networks.