

Friend Recommendation Algorithm Based on Fragmentation of Time and Transmission of Interest

Sun Shibao¹, Jia Bowen¹, Zhang Jingshan¹, Zhang Yanan¹, Yan Xiaolong¹, Yin Lihang¹

¹ College of Information Engineering, Henan University of Science and Technology,
Luoyang, 471000, China

Abstract

In order to solve the problem that the recommendation quality is poor, which results from the fact that the traditional algorithm cannot dynamically generate recommendation for users and collect information, the paper put forward a friend recommendation algorithm which is based on the fragmentation of time and the transmission of interest. Firstly, the paper divides the user's time into several time periods, and calculates the best push time. Secondly, the calculation method of user similarity is redefined based on the Ebbinghaus memory curve and the theory of interest transfer with taking the user browsing time as the standard. Finally, we make experiments by using Gooseeker crawler tools and MATLAB, and the results show that: the calculated similarity mean value and the stability have increased by 38.5% and 33.8%, 2% and 17.1% respectively comparing with the traditional collaborative filtering algorithm and collaborative filtering algorithm based on user similarity.

Keywords: *User Similarity, Transmission of Interest, Friend Recommendation Algorithm.*

1. Introduction

The social network has become an important part in people's life, how to recommend interesting information for users in the social network has Important significance in enhancing uses' sticky and expanding the audience for micro-blog operators. The user's interest varies with time, how to make personalized recommendation based on the changing interests of the users has become the focus of current research. The traditional personalized recommendation method^[1] acquiesces in the assumption that users' interest is relatively stable within a day, rarely considers to provide personalized recommendation service for users with dividing the time into some periods, and the most personalized recommendation algorithm has not been optimized in the push time, and it disturbs users frequently. In the existing algorithm, there many recommendation algorithms classifying users with users' explicit behavior, for example, Xu Jingnan^[2] used the review and scoring method, Dou Lingyuan^[3] used label to divide users. Although these explicit behaviors can intuitively reflect the user preferences of the project in a certain extent, it is not applicable in the use of social networks, because it is

complicated for the users to grade each micro-blog content they have browsed, although it can increase the accuracy of the algorithm to a certain extent, the operation is too complicated that it will lose a large number of users. In the acquisition of users' interest, Xing Xing^[4] presents a time perception recommendation method based on users' feedback, and the method uses the time that the users hesitate to click on the project as the standard time, it has high feasibility, but it highly depends on the location of the project, if the information which a user is interested is at the bottom of the page, the user easily ignores this information when he just opens the page, this extends the time that user hesitates and affects the accuracy of the algorithm. The traditional personalized recommendation system^[5] mainly uses the collaborative filtering methods^[6] to recommend information, and the method takes the similar users as the foundation, first, it finds out the project which similar users have clicked, and then it will recommend the items that the target users have not clicked in this selected set. Collaborative filtering algorithm includes three steps: archiving users, finding the nearest neighbors and recommendation. However, the use of collaborative filtering algorithm is facing the problem of sparse matrix and cold start. In addition, collaborative filtering algorithm mainly takes the similar users to excavates their interests, although the target user and similar users have something in common to a certain extent, the similar user's interest is not the same in the real social network.

Aiming at the above problems, this paper puts forward a new personalized recommendation method. For the instability of interests, the paper puts forward the idea of the fragmentation of time, and thinks that the users have different interests in different time in a single day and makes the optimization in push time. It is tedious to collect explicit feedback, and it highly depends on the location of the project if taking the users' hesitation time as the judgment, in order to solve these problems, the paper takes the implicit feedback information^[7] that how long the users take to browse the information as a measure of user's interest; for the problems that the users have similar attribute but the interest is not necessarily similar, the paper defines a new calculation method of user similarity,

the method not only figures out the contents which the user is interested, but also figures out the similarity of the content they detest. At the same time, in order to be consistent with the actual situation, the paper introduces the Ebbinghaus memory curve to express the forgetting laws of human brain, and according to the actual phenomenon that “love me love my dog”, the paper raised the idea that interests can transmit.

2. Algorithm Description

2.1 The Fragmentation of Time

The fragmentation of time is the main idea for the algorithm design, the traditional personalized recommendation method in social network thinks that user's interest is stable in a certain time, and it will not change in every period of time in a day. But the fact is not the case, for example, although a user loves horror movie very much, he rarely chooses horror movies as the received information before he goes to bed, so it is not appropriate to recommend such information for users at this time. This method has not optimized the push time, often pushes information in people's busy time or rest time, and disturbs the user to rest or work.

Based on the two points above as well as the 2015 annual report on the development of micro-blog users^[8], we found that users are not active from 2am to 10 a.m. Therefore, the algorithm divides the user's behavior into three periods, namely [02:00,09:59], [10:00,17:59]and[18:00,01:59]. In the selection of the push time, we first define the users use time set $User_{Time} = \{User_{Time1}...User_{Timen}\}$, after having obtained the user's using time, we first determine the time point which it is located in the time interval, and set the critical value T_c for the minimum in the interval. The offset time T_p can be calculated by using the following equation.

$$T_p = \frac{\sum_{i=1}^n |UserTime_i - T_c|}{n} \quad (1)$$

Thereinto, n stands for the number of users' behavior, we define the last push time is T_i and $T_i = T_c + T_p$. Now, we set threshold δ , if $UserTime_i - T_c > \delta$, it will not be documented. This setting is designed to avoid the impact of accidental behavior on the push time. The value of threshold δ can be set by the users, the greater the value of δ , the user's abnormal behavior has greater effect on the push time more, contrarily, it has smaller effect.

2.2 The Definition of User's Interest Similarity

The traditional method of friend recommendation only takes into account whether the user has browsed the project when calculating user similarity, and often takes whether the users click on the project to calculate the similarity. In this way, it cannot reject the users who click on the project by mistake, but also ignores the user's preference for the project and results in the deviation of the recommendation quality. Besides, the method only considers the projects that the users have clicked, while ignores projects that the users have not clicked. In the real social network, the aversion projects of users are also an important indicator to evaluate the similarity. According to the above problems, the user similarity is redefined in this paper, and the user's browsing time is taken as the evaluation criterion, the project which the user does not click is also took into account when calculating the similarity.

①The similarity calculation of user's interested content

When users browse the contents they are interested, they often spend more time on the content they are interested very much, for the content they feel a little interested, they use the saccade methods to read and spend less time. Due to the different users have different reading speed, we need to calculate the average browsing speed of users first $Avg_u = \frac{content_u}{time_u}$, $content_u$ stands for the number of words the user has read, $time_u$ stands for the reading time which is spent by users.

Define the similarity of user's interested contents:

$$sim_m(u, u^-) = \frac{\sum_{i=1}^l \Delta\omega \left(\frac{t_{u,i}}{Avg_u} - \frac{t_{u^-,i}}{Avg_{u^-}} \right)}{|I^+(u) \cup I^+(u^-)|} \quad (2)$$

Thereinto, $l = |I^+(u) \cup I^+(u^-)|$ stands for the number of common projects which are browsed by users, and indicates the common preferences between two users. $t_{u,i}$ stands for the time that users spend in browsing project, $t_{u,i} / Avg_u$ stands for the ratio between the time that users take and the average reading speed, and it indicates the user's preference for the current project, while $\left(\frac{t_{u,i}}{Avg_u} - \frac{t_{u^-,i}}{Avg_{u^-}} \right)$ stands for the deviation rate of users' preference for a project.

In order to dynamically represent the change of users' interest, the function of time decay $\Delta\omega$ is introduced. $\Delta\omega = 1 - 0.56(T_1 - T_i)^{0.06}$, it is Ebbinghaus's forgetting function which describes the forgetting law of human brain for information, T_1 is the current time, T_i is the occurrence time of user's behavior, $T_1 - T_i$ stands for the time difference of occurrence time of behavior and the start time, and it is presented by hours.

②The similarity of user aversion content

In the real social network, there are also common things that users do not like and it can generate the aggregation, so the similarity of user aversion content cannot be ignored when calculating users' similarity.

In this paper, the similarity is defined as:

$$sim_{in}(u, u^-) = \frac{|I^-(u) \cap I^-(u^-)|}{|I^-(u) \cup I^-(u^-)|} \quad (3)$$

Thereinto, the algorithm thinks that $|I^-(u) \cup I^-(u^-)|$ is the number of items which are recommended by system, but the users have not browsed it, $|I^-(u) \cap I^-(u^-)|$ is the number of items which are recommended by system, but there are not common users browsing them.

③The similarity of user preference

The user similarity is defined as the sum of the similarity of the user's interest content and the user's aversion content, and α is set as a variable parameter, which is used to reconcile the importance of the two in the algorithm. When $\alpha = 1$, the algorithm only calculated the similarity of user's interested content, at the same time, when $\alpha = 0$, the algorithm only calculated the similarity of user's aversion content. The calculation formula of user preference similarity is as follows:

$$sim_{per}(u, u^-) = \alpha sim_{in}(u, u^-) + (1 - \alpha) sim_{in}(u, u^-) \quad (4)$$

2.3 The transfer of user similarity

Figure 1 is a simple social network attention map, if the user u_1 pays attention to the behavior of user u_2 (it is represented by solid arrows in Figure 1), then the contents which is published or forwarded by u_2 will appear in the list of u_1 , then the interest of u_2 may affect the interest of u_1 imperceptibly (it is represented by dotted arrows in

Figure 1), this is what people often say "Love me, love my dog".

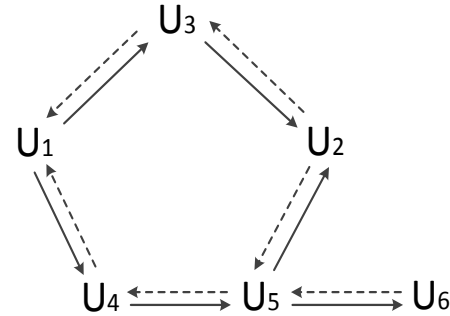


Figure 1 A simple social network attention map

Take Figure 1 as an example, it is considered that when calculating the similarity between the user u_1 and the user u_2 , it not only needs to calculate the similarity of user's preference, the similarity of user's transmission also needs to be calculated. The degree of similarity of transmission interest is defined as:

$$sim_{ps}(u, u^-) = \beta \frac{\sum_{i=1}^l sim_{per}(u, u_i^+)}{R} \quad (5)$$

We define $\beta = \frac{repostCon}{allCon}$, and it indicates that the factors that users are vulnerable to affect by other users, here we take the proportion of forwarding amount among the text they publish as a measure, the more the number of forwarding, the user is more likely to agree with the views of others.

R is defined as the shortest path for user u to get to u^- in a social network, namely the number of hops between two users.

$\sum_{i=1}^l sim_{per}(u, u_i^+)$ stands for the sum of average value of all users and the target user on the path R .

In conclusion, the formula for calculating user similarity is defined as:

$$sim(u, u^-) = sim_{per}(u, u^-) + sim_{ps}(u, u^-) \quad (6)$$

Namely:

$$sim(u, u^-) = \alpha sim_{in}(u, u^-) + (1 - \alpha) sim_{in}(u, u^-) + \beta \frac{\sum_{i=1}^l sim_{per}(u, u_i^+)}{R}$$

Namely:

$$sim(u, u^-) = \left[\alpha \frac{\sum_{i=1}^l \Delta \omega \left(\frac{t_{u,i}}{Avg_u} - \frac{t_{u^-,i}}{Avg_{u^-}} \right)}{|I^+(u) \cup I^+(u^-)|} + (1-\alpha) \frac{|I^-(u) \cap I^-(u^-)|}{|I^-(u) \cup I^-(u^-)|} \right] + \beta \frac{\alpha \frac{\sum_{i=1}^l \Delta \omega \left(\frac{t_{u,i}}{Avg_u} - \frac{t_{u^+,i}}{Avg_{u^+}} \right)}{|I^+(u) \cup I^+(u^+)|} + (1-\alpha) \frac{|I^-(u) \cap I^-(u^+)|}{|I^-(u) \cup I^-(u^+)|}}{R}$$

3. The Implementation Process of Algorithm and Time Complexity Analysis

3.1 Implementation Process

According to the above, this algorithm is divided into two major steps, at the first, segment user's time and calculate the push time for them. Second, calculate the user's interest which is shown in this time period and recommend information for them. Assuming that the algorithm needs to recommend N similar users for the user u , the implementation process of algorithm is as follows:

Input: the current user u , the set of users U , the set of required recommendations user N .

Start:

(1) Calculating the push time

① segmenting time period

According to the time when user has browsed or clicked on the contents, the text is divided into three time periods, and determine the current time, the measure of user's similarity in the subsequent steps is based on the text content in this period of time.

② Calculating offset time

We use $T_p = \frac{\sum_{i=1}^n |UserTime_i - T_c|}{n}$ to calculate the offset time

for user u .

③ Calculating the push time

We use $T_i = T_c + T_p$ to calculate the push time for user u

(2) Calculating user similarity in current time

① Calculating the similarity of user's interest

We use $sim_{in}(u, u^-) = \frac{\sum_{i=1}^l \Delta \omega \left(\frac{t_{u,i}}{Avg_u} - \frac{t_{u^-,i}}{Avg_{u^-}} \right)}{|I^+(u) \cup I^+(u^-)|}$ to calculate

the similarity between user u and other users in the content they are interested.

② Calculating the similarity of user aversion content

We use $sim_{un}(u, u^-) = \frac{|I^-(u) \cap I^-(u^-)|}{|I^-(u) \cup I^-(u^-)|}$ to calculate the

similarity between user u and other users in the offensive content.

③ Calculating the similarity of user interest in the shortest path

We use $sim_{ps}(u, u^-) = \beta \frac{\sum_{i=1}^l sim_{per}(u, u_i^+)}{R}$ to calculate the

similarity between user u and other users in the interest transfer.

④ Calculating the similarity of users

We

use

$$sim(u, u^-) = \left[\alpha \frac{\sum_{i=1}^l \Delta \omega \left(\frac{t_{u,i}}{Avg_u} - \frac{t_{u^-,i}}{Avg_{u^-}} \right)}{|I^+(u) \cup I^+(u^-)|} + (1-\alpha) \frac{|I^-(u) \cap I^-(u^-)|}{|I^-(u) \cup I^-(u^-)|} \right] + \beta \frac{\alpha \frac{\sum_{i=1}^l \Delta \omega \left(\frac{t_{u,i}}{Avg_u} - \frac{t_{u^+,i}}{Avg_{u^+}} \right)}{|I^+(u) \cup I^+(u^+)|} + (1-\alpha) \frac{|I^-(u) \cap I^-(u^+)|}{|I^-(u) \cup I^-(u^+)|}}{R}$$

calculate the general similarity of user u and other users.

(3) Generating the set of recommendation

① Arrange the users in descending order according to the similarity in users set U , and recommend projects for user u .

Output: the recommended set c for the current user, as well as the recommended time T_i for the user

End.

The algorithm calculates the best time to recommend content for user u through the input of user u 's data and user set U , compares the similarity between the user u and each user in the set, and arranges the users in descending order according to the similarity, then it will provide a recommendation set with the number of N for the user u .

3.2 The time complexity analysis of the algorithm

The time complexity analysis of the algorithm is one of the important indexes to measure an algorithm. According to the implementation process of the algorithm in the previous section, the time complexity of the algorithm is analyzed as follows:

```
// calculate the push time
for(int i = 0; i < n; i++){ // n+1 times
    a : the push time // n+1 times
}
// calculate the similarity
for(int j = 0; j < n; j++){ // n+1 times
    for(int l = 0; l < n; l++){ // n(n+1) times
// calculate the similarity of user's interest
    b : the similarity of user's interest // n(n+1) times
// calculate the similarity of user's aversion
    c : the similarity of users' interest // n(n+1) times
// calculate the similarity of user transfer
    for(int k = 0; k < n; k++){ // n^2(n+1) times
        d : the similarity of user transfer // n^2(n+1) times
    }
// calculate the overall similarity of users
    e = b + c + d // n(n+1) times
}
// generate recommendation set
    Engender set // n(n+1) times
}
// recommend N similarity users for the user u
    Carry set out // N times
```

Therefore, the execution time of the algorithm is:

$$n+1+n+1+n+1+n(n+1)+n(n+1)+n(n+1)+n^2(n+1)+n^2(n+1)+n(n+1)+n(n+1)+N$$

Simplify it as: $2n^3 + 7n^2 + 8n + (3 + N)$

Therefore, the time complexity of the algorithm is:

$$T(n) = O(n^3)$$

According to the above analysis, the time complexity of this paper has increased comparing with the previous algorithm, and it will not produce additional burden.

4. Experiment and Simulation

4.1 Experiment Overview

We obtained the information of 50 users of Sina micro-blog by using Gooseeker crawler tool, the information includes age and other attribute information, as well as forwarding content of micro-blog, forwarding time and other non-attribute information. The algorithm also requires the browsing speed and browsing time of users which cannot be obtained by the crawlers, so we assume that the browsing speed of users is same in the algorithm, and use the Radom() function for each valid text to generate random browsing time, and this can ensure the algorithm is effective.

(1) The accuracy analysis of push time

The algorithm calculates the using time and the best push time of 50 users, we only take one user's push time as the example due to space limitations.

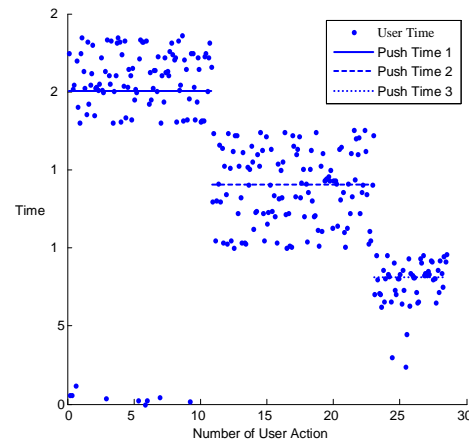


Fig2. Time of user behavior and push results

According to the algorithm, the best push time is 20:04, 14:03 and 8:15 respectively. For the analysis of the experimental results, we take every 30 minute as an interval to record the frequency of the time points. We think that the frequency of users' using time in the interval is greater, it is more unlikely to disturb the users if pushing content for them.

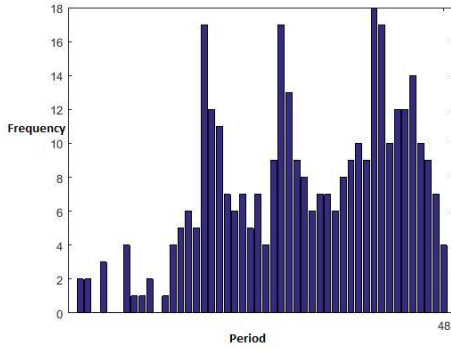


Fig3. The frequency distribution of users' behavior

According to the above results, the push time is located in the period of highest frequency of user's history point, indicating that the user in this time point is more likely to be at leisure, if pushing content at this time, it is not likely to disturb the users.

(2) The analysis of variable parameter's value

In this paper, a variable parameter is used to adjust the preference information and the aversion information. In this experiment, we calculate the similarity of the target user and the random 20 users when setting the value of α is 0.32, 0.42 and 0.52, and we also discuss the effect of α on the similarity calculation, the results are as follows.

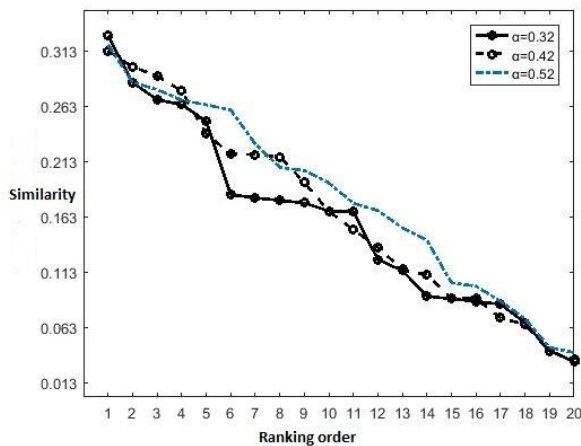


Fig. 4 The influence of α on users' similarity

The paper uses the standard deviation
$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$
 to measure the dispersion degree of numerical value, thereinto, N is the number of

users, x_i is the similarity between each user and the target user, μ represents the mean value of similarity.

When $\alpha = 0.32$, $\sigma = 0.08336$; when $\alpha = 0.42$, $\sigma = 0.087123$; when $\alpha = 0.52$, $\sigma = 0.08218$. According to the experimental results, we can see that when $\alpha = 0.52$, the σ and the standard deviation are the smallest among the three, and the algorithm is the most stable.

(3) The contrastive analysis of the algorithm's accuracy

According to the preamble, when $\alpha = 0.52$, the algorithm is the most stable. In order to take the contrastive analysis of the algorithms, we choose the traditional collaborative filtering algorithm (CF), the collaborative filtering algorithm (U-CF) which is based on the user similarity and the algorithm (F&U-CF) of this paper to provide recommendation list for the user "Cheng tian xiang shang" in the data set. The results are shown in the following figure 5:

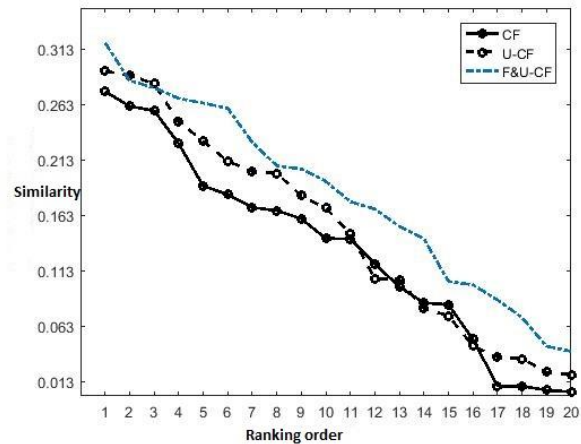


Fig.5 The contrastive analysis of several algorithms' similarity

The algorithm of this paper calculates the similarity mean value of recommended list for the target user $\mu = 0.179696$, the standard deviation is $\sigma = 0.08218$. The traditional collaborative filtering algorithm calculates the similarity mean value is $\mu = 0.129691$, the standard deviation is $\sigma = 0.08458$. The collaborative filtering recommendation algorithm based on user similarity calculates the similarity mean value is $\mu = 0.134268$, the standard deviation is $\sigma = 0.096283$.

Therefore, the algorithm of the paper has obvious improvement in terms of similarity mean and stability comparing with the above two methods. In this paper, the mean value of the algorithm has improved by 38.5% and 33.8% respectively compared with the first two methods,

and the stability has improved by 2% and 17.1%, respectively.

5. Conclusion

The paper divides the user's time into several time periods, optimizes the push time and improves the push quality. The calculating method of user similarity has been redefined in the social network, and the user's browsing time is taken as the standard, besides, the paper calculates the similarity of the content which user is interested as well as the content they are not interested. In order to show the change of user's interest dynamically, the paper uses the Ebbinghaus memory curve to simulate the forgetting rules of human brain. In order to accord with the actual situation that "Love me, love my dog", we also introduce the transmission mechanism of interest.

The similarity mean and the stability of the algorithm in this paper have increased by 38.5% and 33.8%, 2% and 17.1% respectively comparing with the traditional collaborative filtering algorithm and collaborative filtering recommendation algorithm based on user similarity, so it can be widely used to release personalized recommendation in the social network, push mobile phone APP applications, and release personalized recommendation of e-commerce website etc. The algorithm only considers the text information in the social network, while multimedia information is an important part in the real social network, next we will focus on this part.

Acknowledgments

This work is partially supported by Key Projects of Henan Province (152102210277), Research Cooperation Projects of Henan Province (152107000027), Program for Innovative Research Team (in Science and Technology) in University of Henan Province (17IRTSTHN010), Technological Innovation Team of Henan University of Science and Technology (2015XTD011), Major Cooperative Engagement Fund of Henan University of Science and Technology (2015ZDCXY03).

References

- [1] Gao Ming, JIN Cheqing, QIAN Weining, et al. Real-Time and Personalized Recommendation on Microblogging Systems[J]. Chinese Journal of Computers, 2014, 37(4): 963-975.
- [2] Xu Jingnan, "Research on Personalized Recommendation Algorithm Based on Comments and Scores", M.S.thesis, Department of Computer Science and Technology, Zhejiang University, Zhejiang, China, 2013.

- [3] Dou Lingyuan, Wang Xinhua. Collaborative Filtering Recommendation Algorithm Based on Time and Label Context [J]. Journal of Taiyuan University of Technology, 2015, 46(6): 735-748.
- [4] XING Xing, JIA Zzhichun, YANG Yuqiang. User Feedback Based Method for Time-Aware Recommendations[J]. Computer Engineering and Applications, 2015, 51(22):141-144.
- [5] WANG Zhisheng, LI Qi, WANG Jing, et al. Real-Time and Personalized Recommendation on Implicit User Feedback Data Stream[J]. Chinese Journal of Computers, 2016, 39(1): 52-64.
- [6] SUN Haifeng, GAN Mingxin, LIU Xin, et al. Review on Dominating Websites for Movie Recommender Systems[J]. Journal of Computer Applications, 2013, 32(S2): 119-124.
- [7] Yang Changcun, Yang Jing, Ding Hong. A New Friends Recommendation Algorithm for Sian Microblogging[J]. Computer Applications and Software, 2014, 31(7): 255-274.
- [8] knowledge base. 2015 microblogging user development report[EB/OL]. [2016-02-18]. <http://www.useit.com.cn/thread-10921-1-1.html>.

First Author Born in 1970, Ph.D., professor. His research interests include intelligent information processing, digital image processing.

Second Author Born in 1992, M. S. candidate. His research interests include intelligent information processing.

Third Author Born in 1992, M. S. candidate. His research interests include computer network, online public opinion.