

Improved Recommendation System Using Friend Relationship in SNS

Qing Liao^{1,3(✉)}, Bin Wang^{1,3}, Yanxiang Ling^{1,3}, Jingling Zhao², and Xinyue Qiu²

¹ School of Information and Communication Engineering,

Beijing University of Posts and Telecommunications, Beijing 100876, China

² School of Computer Science, Beijing University of Posts and Telecommunications,
Beijing 100876, China

jing_lingzh@sina.com, 365562443@qq.com

³ Beijing Key Laboratory of Network System Architecture and Convergence,
Beijing 100876, China

qingliao@hotmail.com, wangbinjs@126.com, cdxustb@163.com

Abstract. With the rapid development of the Internet, SNS services and 3G commercial mobile applications there have been tremendous opportunities although the development of SNS is very short in China, and the social web game is in the early stage of development. Because of massive users, the potential commercial value of Chinese SNS is still a great mining space. However, a relatively large defects is the precipitation and accumulation on content. The dynamic of friends will affect our own decisions largely, it is favorable for the activity of SNS to increase the number of friends. We have improved the existing models, and conduct experiments to validate it and compare it with previous methods.

Keywords: SNS · Context · Matrix decomposition model · Recommendation

1 Introduction

Since the birth of the search engines Google, we are discussing what is the next gold mine of the Internet. Now, almost everyone agrees that it is the social network. According to the report, Internet users almost take 22 % of the time in the social media and social networks. Facebook and Twitter as a representative of two different types of social networks, are pioneers in the Internet sector SNS. Social networks have two most important factors: social relationships and social data information, social relationships also known as friendships. As we all known, dynamics of our friends on the SNS will largely affect our own decisions. For example, we show their new shoes in the SNS, it is likely to affect our friends. This kind of influence is greater than the power of the seller. That is to say, friends can spread the “trust”. Based on this, the SNS pay more attention to the topology relationship of the user’s friends.

According to the findings of the U.S. investigation agency Nielsen, it shows that more than 90 % of users believe in what their friends have recommended, and then 70 % of users believe the score of the product that other users rated on a commodity in the Internet. Therefore, what friends recommend is important to increase user’s trust recommendation result. Nielsen once made a personalized ads experiments with Facebook,

and do the ABtest. Nelson display the same brand in three different ways of advertising, the first is to tell the users of Facebook that there are 50000 users pay attention to the brand, the second is to tell the users of Facebook that how many friends pay attention to this brand, and the third is to tell users what friends pay attention to the brand. According to the analysis of AD click results, the effect of the third is better than that of the second, the second is better than the first. So increasing the number of friends is conducive to increase the activity of the social product, which increases the effect of advertising.

2 Recommended Types of Social Networks

2.1 The Representation and the Characteristics of Social Network Data

Social networks (SNS) define the relationship between users, according to the definition of figure, we represent SNS by $G(V,E,W)$, and V represents the vertices, E represents the set of edges. If the user V_a and V_b have social relationships (which is often said friends relationship), then there is an edge $e(V_a,V_b)$ connecting these two users. $W(V_a,V_b)$ represent the weight of edge which shows the degree of relationship between friends, and most can be set to 1. Today there are two kinds of SNS, one is Facebook that represents the bidirectional relationship of friends, $e(V_a,V_b)$ which is an undirected edges can describes it. Another is twitter that represents the unidirectional relationship between friends, so directed edges represent the relationships.

$Out(u)$ is a set of vertices which vertex u point to in the Figure G , $in(u)$ is a set of vertices which point to vertex u . Generally, there are three different social network data:

- (1) SNS of bidirectional relationship of friends; for example, Facebook and renren, $out(u) = in(u)$.
- (2) SNS of unidirectional relationship of friends; the typical representative is Twitter and Sina weibo.
- (3) SNS of community groups, the relationship between SNS users is uncertain, however, this data includes the user data belonging to different communities. For example, douban group, which belonging to the same group may represent the users have similar interest.

There are two types of recommendations based on social networks, one is recommended items, items can be advertising, news and more. Another one is friend's recommendation.

2.2 Recommended Items Based on Social Networks

Many websites recommend products to users by SNS, for example, Amazon recommends product to users based on what their friends like.

Recommendation from friends can increase the recommendation trust. Friends of the users on the SNS tend to be trust, at the same time the users are disgusted with the results of computer calculated. For example, Web site recommend "naruto" to users, if it is based on the item with the filtering algorithm, the recommended reason may be that

“naruto” and “one piece” is similar. But if based on SNS recommended, the recommended reason is that there are 10 friends of the user like “naruto”, apparently users tend to accept the second reason.

Ease the problem of cold starting. When a user login weibo on e-commerce sites, we can get the user’s list of friends from weibo, then get the consumption records of their friends’ and find out the right item to recommend to the current user.

Of course, the method of item recommendation based on SNS also can produce bad case. Because the user’s friends relationship are not usually based on common interests (relatives, classmates friends are also friends). So the different interest of user’s friends, make recommendation accuracy and recall rate of the algorithm decreased.

The behavior of users that buy goods can be shown by users diagram and Items diagram, and the combination of the figure and users of social network diagrams together is a bipartite graph. Figure 1 is an example of a bipartite graph combines social networking and user objects. If a user u produce a behavior for goods i , then there is the edge between two nodes. For example, a user buy objects a and objects e , If user B and D are friends, then there will be an edge connecting the two users, the user A in Fig. 1 and the user B , D are friends.

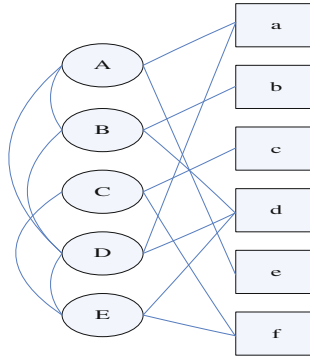


Fig. 1. The combine of social networking and user objects graph

The similarity between users also means the weight between users, which can be defined as α . The weight between the user and the items may be defined as the degree of the product user liked, which can be defined as β . The value of α and β are based on the training data. After determining the edge weight, we can use PersonalRank figure sorting algorithms to generate recommendations for each user.

In a social network, in addition to the common and directly relationship between the user’s social network, there is another relationship which the user belong to the same group (such as douban interest group). At this point, the map mode will be more simple. We can modeling the relationship between the group and the user’s friends. Figure 2, adding a node that represents a community, and if the user belongs to a community, there is an edge between the group nodes and the user nodes.

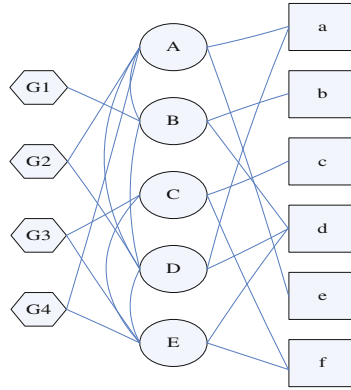


Fig. 2. Integration of two social network graph model information

2.3 Recommend of SNS Friends

The relationship of good friends is the important part in the SNS, if the user's friends are very few, it will not be able to realize the benefits of socialization. Therefore, recommending friends is one of the important application of SNS. The recommended system of friends is to recommend product by friends and the interact daily record of user's behavior, first briefly introduce some simple algorithm, and put forward our own recommendation algorithm of SNS friends.

(1) the matching algorithm based on content

User content has many types [5].

- a. User Demographics attribute content, which includes age, sex, work, residence, education and so on.
- b. User interest which is the most important content, including user favorite items and published remarks.
- c. User context, including GPS location, mood, weather, time, etc.

Based on the above information, we calculate the similarity between users, and then recommend friends. In fact, these three pieces of information called user context information. In the proposed algorithm, the full use of context information can improve the accuracy of the recommended.

(2) recommendations based on common interest of friends

In SNS friends, the users become a friend based on the common interest, they do not care if they knew each other in the real world. Therefore, recommendations based on common interests of friends is needed in sina microblogging.

User-based collaborative filtering algorithms can be used to calculate the user similarity, the main similarity analysis is based on the same score on the same items, for example, when users comment or forwarding the same microblogging, then we believe that user likes this microblogging.

- (3) Recommendations of friends based on a social network graph.

We analysis social networking by the graph model, especially new users, we recommend friend by user groups, or recommend friends of friends by taking advantage of the spread of the map, here is a recommendation algorithm of friends based on social networking.

We can calculate their similarity degree based on the proportion of common friends for user U and V:

$$w_{out}(u, v) = \frac{|out(u) \cap out(v)|}{\sqrt{|out(u)| |out(v)|}} \quad (1)$$

out(u) means a set of friends collections that user U point to, in(u) means a set of friends collections who point to user U, approximation defined as in(u) can be represent as follow:

$$w_{in}(u, v) = \frac{|in(u) \cap in(v)|}{\sqrt{|in(u)| |in(v)|}}. \quad (2)$$

3 Improved Recommendation of SNS Friends: Matrix Decomposition Model on Context

As is known to all, SNS is a gold mine, especially the rapid development in recent years, as the representative of SNS, sina Weibo and tencent weibo daily output plenty of information in China. So in order to get more accurate information and people, to build a good friend recommendation system is indispensable. Without the users' activity, SNS service is a "dead city". Traditional recommendation systems, such as user-based CF and Item-based CF, both only taking the user or objects into account, but ignoring the property of users or items. As a result, it will be difficult to grasp the true preferences of the user, and the recommendation accuracy can not be guaranteed. Therefore, we must fully consider the context to improve accuracy of recommendation. This paper will present an improved algorithm and model, which can improve the accuracy of recommendation with the integration of the context of the user's, the context of time and the

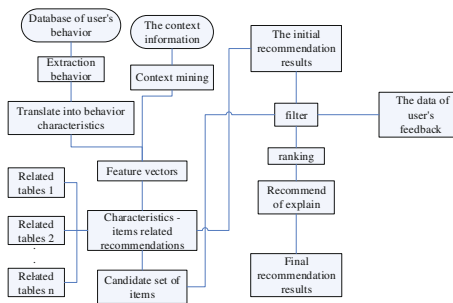


Fig. 3. Recommend recommendation system model based on context

matrix decomposition model. Figure 3 is our recommendation system model based on context.

Simply explain the role of the various modules

- a. The Database of user behavior. Users can create many different kinds of behaviors, such as user browse items, click on the link address of items, add items to shopping cart, purchase items, to score, etc. These behaviors can make guidance to the recommend of users. Each kind of behavior can be used as a dimension of user behavior feature vector, through training and learning, and get the final weight of behavior.
- b. The context information. A good recommendation system, must consider the context, this context can be the user's population information, attributes of the item, the time context, site context, etc. But in some cases, certain context is not obvious, for example, we only know the IP address of the user, in order to get the user's location, you need the mining module to analysis the context.
- c. Related tables. Related tables can be understood as a configuration file, stored the weight on a characteristic value.
- d. Characteristics - items related recommendations. After the feature vector of the customers, according to the configuration of related tables, we get the initial list of recommended items offline.
- e. Filter module. To filter out the following items:
 - (1) The items which the user has behaved.
 - (2) The items besides candidates. Such as user search clothes in taobao, rules must be men's clothes, you need to get rid of the clothes which are not men's.
 - (3) Poor quality items. To get a lot of bad review, you need to get rid of these products.
- f. Ranking module. Ranking module is to ensure the novelty of the item, which would require to drop the weight of popular items. In addition, need to ensure diversity, and this need to cover the user's various interest. Finally, to guarantee time diversity, meaning that the user does not want to see the same recommendation every day, so the recommendation system should focus on recent actions of users.
- g. User feedback. User feedback can help recommend system to get the user's interest. We can predict that whether the user can click the recommendation results based on the click models.

3.1 The Introduction of Matrix Decomposition Model

The essentially of Matrix decomposition model is to find the potential characteristics of users, in the field of text mining, also known as latent factor model (LFM) [1–4]. Since the Netflix Prize, the LFM has gradually become the powerful tool of the recommendation system. For the User - based CF, you first need to find users who read the same books with the recommended users, and then recommend other books which that users liked. For the Item-based CF, we need to recommend the book which is similar to that books they have read. First of all, classify items by characteristics, for users, get his interest categories, then we select items he might be interested. Classification can be classified by the editorial staff, but it is difficult for us to determine the right of an item.

And because matrix decomposition model take use of automatic clustering based on the statistics of user's behavior, it becomes a better solution to the problem of classification weights.

(1) Singular value decomposition

Recommendation system can extract a set of potential (hidden) factors from the rating model by using matrix factorization, and describe users and items by these factors vectors. In the late 1980 s, using the ideas of potential semantic lots factor has been successfully applied in the field of information retrieval. Deerwester proposed using singular value decomposition (SVD) technique to find potential factors in the document. In information retrieval, this kind of latent semantic analysis (LSA) technology is also classified as latent semantic retrieval (LSI).

Problems in the field of information retrieval are usually based on the user's query words to find a set of documents. Document and the user query is resolved as a vector of words. This retrieval method can't solve the synonyms in the document or the query words. SVD regards that highly relevant and appear together words as individual factor, and make the usually large document vector matrix apart into smaller order approximation matrix. So we can retrieve relevant documents in the case that do not contain user query words based on LSI.

In the field of recommender systems, people pay more attention to the latent semantic model and matrix decomposition model. In fact, the nature is dimension reduction techniques, and score matrix completion.

User's scores on items can be expressed as a rating score matrix R , and $R[u][i]$ indicates that the user's scores on item i . However, the user can not comment on all items, so many elements of this matrix is empty, such as if a user does not score on an item, then they would have to predict whether the user is able to score for this item and assess how many points.

For how to complete a matrix, there are a lot of research in the history. There are many ways to complete an empty matrix. We want to find a completion of matrix perturbation method which have a minimum disturbance. So which is the minimum disturbance for matrix? It is generally believed that if the eigenvalues of the matrix after completion and before the completion of eigenvalues were similar, we said that the disturbance for matrix is minimum.

The first matrix decomposition model is SVD [6], there are m users and n items, and scoring matrix $R^{m \times n}$. First we need to complement the missing items' score in the matrix, by the way of using the global average or the average of users and items, then get the completion matrix R' . Then we decompose R' into the following form:

$$R' = U^T S V \quad (3)$$

$U \in M^{k \times m}$ and $V \in M^{k \times n}$ are two orthogonal matrix, $S \in M^{k \times k}$ is a diagonal matrix, each of the diagonal elements are the singular values. In order to reduce the dimension of the R' , we take the largest singular values and composed the diagonal matrix S_f , and get a scoring matrix with dimensionality reduction:

$$R'_f = U_f^T S_f V \quad (4)$$

SVD is used by the early recommendation system, but this method has the disadvantage:

- (1) This method first need to complement the sparse scoring matrix with a simple method. So matrix will become a dense matrix storage, and take the great storage space.
- (2) The calculation of this method is large, especially in a dense matrix. The actual recommendation system is on the hundreds of millions of users and even one hundred million level of goods, apparently this method can not be used.

We'll decompose the score matrix R into a low latitudes matrix multiplication:

$$\hat{R} = P^T Q \quad (5)$$

$P \in M^{f \times m}$ and $Q \in M^{f \times n}$ are two dimension reduction matrix. P represents user implicit characteristic vector, Q represents the hidden feature vector of items, f represents the number of users' or items' hidden features, we can define our own, in theory, the greater the accuracy is higher, but the memory consumption space is greater. $\hat{R}(u, i)$ means the score of the item i from user u .

$$\hat{R}(u, i) = \hat{r}_{ui} = \sum_f p_u q \quad (6)$$

Loss function is as followed:

$$\begin{aligned} C(p, q) &= \sum_{(u,i) \in \text{train}} (\hat{r}_{ui} - r_{ui})^2 \\ &= \sum_{(u,i) \in \text{train}} (r_{ui} - \sum_{f=1}^F p_{uf} q_{if})^2 \end{aligned} \quad (7)$$

Then we add prevent fitting items $\lambda (\|p_u\|^2 + \|q_i\|^2)$, so the formula is as followed:

$$\begin{aligned} C(p, q) &= \sum_{(u,i) \in \text{train}} (\hat{r}_{ui} - r_{ui})^2 \\ &= \sum_{(u,i) \in \text{train}} (r_{ui} - \sum_{f=1}^F p_{uf} q_{if})^2 + \lambda (\|p_u\|^2 + \|q_i\|^2) \end{aligned} \quad (8)$$

Then we optimize parameters with stochastic gradient descent algorithm, SGD through continuous iteration strategy, reduce the iteration error, stopped until the compressed to a certain range of allowable error.

$$\begin{aligned}\frac{\partial C}{\partial q_{if}} &= -2p_{uk} + 2\lambda q_{ik} \\ \frac{\partial C}{\partial p_{uf}} &= -2q_{ik} + 2\lambda p_{uk}\end{aligned}\quad (9)$$

The recursive formula is as followed:

$$\begin{aligned}p_{uf} &= p_{uf} + \alpha (q_{ik} - \lambda p_{uk}) \\ q_{if} &= q_{if} + \alpha (p_{uk} - \lambda q_{ik})\end{aligned}\quad (10)$$

α is the learning rate, it needs to get through trial. A scoring system has some inherent attributes which have nothing to do with users and items, and some properties of the items also has nothing to do with the users. So, someone put forward the following matrix decomposition model, the prediction formula is:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T p_i \quad (11)$$

μ : The global average of all the records of score of the training sets. In different sites, because the site location and items are different, the distribution of the overall site's grade will also show some differences. Global average can represent the impact on user ratings with the website itself.

b_u : User bias. It represents the unrelated factors with items in the user's rating habits. For example, some users are demanding, some users are more tolerant, then it will appear different ratings for the same items.

b_i : Item bias. It represents the irrelevant factors with users in the ratings of items. For example, some items itself has high quality, so the score is relatively high, and poor quality of some items, relative score will be low.

Modified ItemCF prediction algorithm is as follows:

$$\hat{r}_{ui} = \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} w_{ij} \quad (12)$$

w_{ij} can be optimized by the following optimizing loss function:

$$C(w) = \sum_{(u,i) \in T} \min \left(r_{ui} - \sum_{j \in N(u)} w_{ij} r_{uj} \right)^2 + \lambda w_{ij}^2 \quad (13)$$

However, w matrix of the model is very dense, for storage, it will cost a lot of memory space. In addition, if there are n items, then the parameters of the model number is n^2 , the number of parameters is large, so it is easy to cause the fitting. Koren[8] proposed that we should decompose the w matrix, the parameter number decreased to $2 * n * F$, model is as follows:

$$\begin{aligned}
\hat{r}_{ui} &= \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} x_i^T y_j \\
&= \frac{1}{\sqrt{|N(u)|}} x_i^T \sum_{j \in N(u)} y_j
\end{aligned} \tag{14}$$

We will add the LFM and the above model, so as to get the following models:

$$\begin{aligned}
\hat{r}_{ui} &= \mu + b_u + b_i + p_u^T q_i \\
&\quad + \frac{1}{\sqrt{|N(u)|}} x_i^T \sum_{j \in N(u)} y_j
\end{aligned} \tag{15}$$

Koren in order not to increase too much parameter to fitting, we define $x = q$

$$\begin{aligned}
\hat{r}_{ui} &= \mu + b_u + b_i \\
&\quad + q_i^T \left(p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j \right).
\end{aligned} \tag{16}$$

3.2 Improved Model

Compared with traditional recommendation problem, recommend friends for weibo is more challenging. Weibo users have rich interaction records. Such as tweeting, forwarding, commentary, attention/concern, @, each kind of interaction can be regarded as some preferences of users. Moreover, the recommended item is a special kind of users, and have the same behavior with ordinary users. The most important, the environment context and time context of users also determine the quality of friends recommend, user context can be the attributes of users, such as age, occupation, company, etc. while the time context will take into account the currently preferences of the users, because the preferences will change with time, so the time context is essential.

U : User sets, u means one of the users, $|U|$: the number of users.

I : Items set, I means one of the items, $|I|$: the number of the recommend item.

r_{ui} : It represents the score of items from users, for weibo friend recommendation, only two points, 1 means accepted and 0 means not accepted.

\hat{r}_{ui} : It represents predict score of items from users.

$R_{m \times n}$: It represents rating matrix of n items from m users.

Firstly the user's interest will change with time, for example, he likes football today, however he maybe interested in basketball in the next days. Secondly the user's behavior on SNS will be different during the day and night, for example, a user is working during the day, so because of time, he is likely to browse weibo quickly, but seldom pay attention to the recommended friends, on the contrary, the evening will have a lot of time to deal with weibo for details. Thirdly the popularity of the article may be reduced with time. Based on the above, this article will integrate the context in the LFM model.

We add the user's context into LFM model:

$$\hat{r}_{ui} = \mu + b_{ui}(t) \tilde{q}_i^T \tilde{p}_u \quad (17)$$

μ represents the global average score of all the records in the training sets, in different sites, because the different site location and different items, the overall site grade distribution will also show some differences. Global average can make impact on user ratings of the website itself. $b_{ui}(t)$ represents the offset item of users and items. $b_{ui}(t) = b_u(t) + b_i(t)$, $b_u(t)$ and $b_i(t)$ are the bias of the context, \tilde{q}_i and \tilde{p}_u represents the model of items and the user.

- (1) The attributes of users and items integrate into the context of the article.

We can dig out some users' interests and tastes with the context of the attributes of users and objects, such as age, sex, job, living area and so on. Users with similar age generally have similar hobbies, so we added two pairs of bias items: $b_{u,gen(i)}$, $b_{u,age(i)}$, $b_{gen(u),i}$, $b_{age(u),i}$. $b_{u,gen(i)}$, $b_{u,age(i)}$ indicates the preferences bias of recommended friends who has similar age and same sex, $b_{gen(u),i}$, $b_{age(u),i}$ represents preferences bias on users by some people who have a similar age or sex. Assuming the user's registered birth year is y , then do the following age categories:

$$age(u) = \begin{cases} 0 & y \leq 1949 \\ \text{ceil}\left(\frac{y-1950}{3} + 1\right) & 1950 \leq y \leq 2004 \\ 16 & y \geq 2004 \\ 17 & y \text{ is illegal} \end{cases} \quad (18)$$

Sex categories:

$$gen(u) = \begin{cases} 0 & \text{if sex is male} \\ 1 & \text{if sex is female} \\ 2 & \text{if sex is null} \end{cases} \quad (19)$$

- (2) The user activity bias into the context of the article.

User activity means the number of microblogging that user send or forward, the higher the number, the more its activity. The bias degree of user's active is $b_{twnum(u)}$ $twnum(u) = \text{ceil}(\text{numTweet}(u))/5 + 1$ means the number that user send.

- (3) The context of user tags.

Tags marked users and the characteristics of recommendation, for example, a user labeled themselves as "IT", "Football", then we can make sure that this user is very interested in several of these field, so we can know the similarity between users with the similarity of tags. $\text{taglist}(u)$ is the tag list of user u and $\text{taglist}(v)$ is the tag list of objects that to be recommended. Then the similarity between them is expressed as:

$$sim_{tag}(u, v) = \frac{taglist(u) \cap taglist(v)}{taglist(u) \cup taglist(v)} \quad (20)$$

The tag's bias of user u and user v is expressed as follow:

$$b_{tag} = \alpha_{u,v} sim_{tag}(u, v) \quad (21)$$

Then we define the variable:

$$q_{u,tag} = \frac{\sum_{n \in T(u)} Vec(n)}{|T(u)|} \quad (22)$$

$$q_{i,tag} = \frac{\sum_{n \in T(i)} Vec(n)}{|T(i)|} \quad (23)$$

$T(u)$ means the tag collection of user u , $Vec(n)$ means n vectors of the tag.

- (4) The content of microblogging and keywords.

When a user forwards or write weibo, essentially reflects the potential interest of the user. So we extract microblogging keywords, and use the similarity of keyword to analyze the association of users and items. Keywords can be extracted from the microblogging, forwarding microblogging, comments or the user's self-description. There is a weight for keywords k from each user means $w(u,k)$.

Similarly, If a user u and item i have the same keywords, the keywords can be seen the common interest of both, we can get a formula which is similar to the bias of tag items. The tag list of user u is $kwlist(u)$ and the tag list of objects that to be recommended is $kwlist(v)$, then the similarity can be expressed: $sim_{kw}(u, v)$

We define the offset of keywords u and keywords v :

$$b_{kw} = \beta_{uv} sim_{kw}(u, v) \quad (24)$$

β_{uv} is the weight of $sim_{kw}(u, v)$.

- (5) The time integrate into the context.

The popularity of the recommended items will change over time. For example, a movie, it is possible becomes popular with the factors that from outside. So for the LFM model, items bias b_i can no longer be assumed to be constant term, while it should be a time-related functions. In addition, users bias b_u should be a time-related functions. We defines the user - Item Bias items:

$$b_{ui}(t) = b_i(t) + b_u(t) \quad (25)$$

$b_{ui}(t)$ is the bias of items i from user u at time t , $b_u(t)$, $b_i(t)$ is the time offset function of users and items.

We discrete the time, each day is divided into time periods, each time period is one hour, $hour(t)$ represents the number of hours a day (from 0 to 23). $day(t)$ is the date in days (from 0, the maximum value of $day(t)$ is 29). So the bias function of items is:

$$b_i(t) = b_i + b_{i,day(t)} + b_{i,hour(t)} \quad (26)$$

Based on the feedback time on recommendation from user, we introduce the time decay function which represent the degree of influence on scores:

$$c_{u,t} = \frac{1}{1 + \alpha_u \left(\frac{t-T_b}{T_e-T_b} \right)^2} \quad (27)$$

$c_{u,t}$ is the degree of influence on scores from users at time t , T_b is the start timestamp of training data, t is the current timestamp, T_e is the end timestamp of training data, α_u is the attenuation coefficient.

$$b_u(t) = b_u + \frac{1}{1 + \alpha_u \left(\frac{t-T_b}{T_e-T_b} \right)^2}. \quad (28)$$

4 Experiment and Analysis

4.1 A Set of Training Data

The date is from tencent microblogging, the data is sampled from four hundred million tencent microblogging users about fifty days, which contains more than two hundred million active users, six thousand recommended users or information, more than three hundred million records and listening anction, more than seventy million training records, and more than thirty million test records.

4.2 Evaluation Standards

We suppose to recommend an ordered list of m items to a certain user, the user may select one of 0, 1 or more of the recommended items, then we define the average accuracy from information retrieval. If the denominator is zero, the result is set to 0. $P(k)$ represents the accuracy at the cutoff point k on the list of items. When the k -th item is not selected, $P(k)$ is zero. Here are a few examples:

- (1) If the user selects the # 1, # 3, # 4 from the five recommended projects, its accuracy is

$$ap@3 = (1/1 + 2/3)/3 \approx 0.56$$

- (2) If the user selects the # 1, # 2, # 4 from the four recommended projects, its accuracy is

$$ap@3 = (1/1 + 2/2)/3 \approx 0.67$$

- (3) If the user selects the # 1, # 3 from the three recommended projects, its accuracy is

$$ap@3 = (1/1 + 2/3)/2 \approx 0.83$$

For the N users in the position n , the average accuracy is the accuracy of the mean average of each user, the following equation:

$$AP@n = \sum_{i=1,2,\dots,N} ap_i@n_i/N. \quad (29)$$

4.3 The Results of Experiment

Table 1 shows the results of the recommendation system based on the context in the SNS:

Table 1. The results of the recommendation system based on the context in the SNS

type	describe	MAP@3
1	A	0.3443
2	1+ age	0.3606
3	2+ item	0.3611
4	3+ microblog	0.3621
5	4+ time	0.3697

A means the Basic matrix decomposition model.

Figure 4 shows the trends of change in the recommended results:

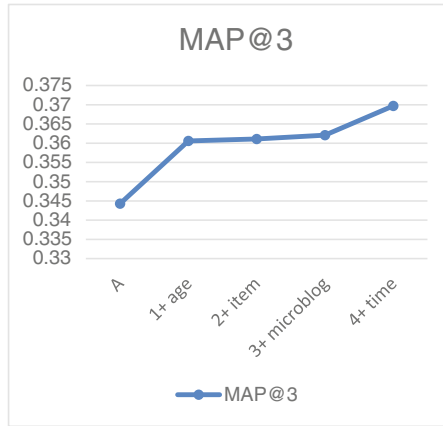


Fig. 4. The trends of change in the recommended results

As can be seen from Fig. 4, add the bias of time context into the system, it can effectively improve the recommendation accuracy.

5 Summary

Social networking can be the current gold mine. When users has more and more friends in the social network, SNS will be more active, which will bring more profits for social networking. Therefore recommending friends for users of SNS is also one of the important functions of the SNS. This artical apply the time context to the bias model of users and items, and excavate the context of user's activity. We extended matrix decomposition model with the demographic attributes of users and items, and verify the model with the data of tencent microblogging, at last achieved a good results on recommendation.

References

1. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 426–434 (2008)
2. Koren, Y.: Factor in the neighbors: scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data (TKDD)* **4**(1), 1 (2010)
3. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
4. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 263–272. IEEE (2008)
5. Hong, J., Suh, E.H., Kim, J., et al.: Context-aware system for proactive personalized service based on context history. *Expert Syst. Appl.* **36**(4), 7448–7457 (2009)
6. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: *Recommender Systems Handbook*, pp. 217–253. Springer US (2011)
7. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the Fourth ACM Conference on Recommender systems, pp. 135–142. ACM (2010)
8. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)

Transactions on Computational Collective Intelligence
XIX

Nguyen, N.T.; Kowalczyk, R.; Xhafa, F. (Eds.)

2015, XI, 187 p. 65 illus. in color., Softcover

ISBN: 978-3-662-49016-7