ISSN : 0974 - 7435 Volume 10 Issue 9

2014

BioTechnology

An Indian Journal

FULL PAPER

BTAIJ, 10(9), 2014 [3846-3855]

User integrated similarity based collaborative filtering

Tian-Shi Liu, Nan-Jun Sun, Liu-Mei Zhang
School of Computer Science, Xi'an Shiyou University, Xi'an 710065, (CHINA)
E-mail : liutianshi@xsyu.edu.cn; sun_coke007@163.com

ABSTRACT

Traditional similarity calculation method in collaborative filtering is inaccuracy due to the extreme sparsity of user rating data. To address this problem, we propose a collaborative filtering recommendation algorithm based on user integrated similarity. The algorithm modifies the similarity calculation formula by introducing the common factor. Then it introduces the item category interestingness eigenvector by category of items and distribution of user ratings to construct the user's item category interestingness similarity. Finally, it combines the user rating similarity to construct the integrated similarity, and generates recommendations. The experimental results show that this algorithm can effectively relieve the inaccuracy of traditional similarity calculation method in the case of extreme sparsity of user rating data, and improve the quality of the recommendation of recommender systems.

KEYWORDS

Recommender systems; Collaborative filtering; User integrated similarity; Item category interestingness eigenvector; Data sparsity.

INTRODUCTION

With the rapid development of internet information technology, the problem of information overload is increasingly prominent, making it difficult for people who faced with a huge mass of data to get the real useful parts they like. Therefore, personalized recommender systems have been proposed. Personalized recommender systems recommend information and commodities to users according to the interests and buying behaviors of users. The nearest neighbor collaborative filtering (CF) has already been the most successful recommendation technology in personalized recommender systems[1-4].

However, due to the increasing number of users and commodities, the problems of data sparsity[5], cold start[6] and scalability[7] in recommender systems are severer, directly affect the recommendation quality of recommender systems. In order to solve these problems, researchers has proposed several new methods. For instance, paper[8] proposed that combining the collaborative filtering recommendation with the content-based recommendation can effectively alleviate the impact of cold start; paper[9] proposed that to alleviate the problem of sparsity, methods such as Default Voting, Inverse User Frequency can be used.

Although, the methods above, to some degree, are able to reduce the sparsity of data and disadvantages of traditional similarity calculation from different angles, but they are not able to utilize the item category information to get user's interests accurately, thus leading to the unsatisfactory quality of recommendation. In allusion to the problems mentioned above, in this paper, we propose an improved user-based collaborative filtering algorithm. The algorithm takes advantage of category information of items to classify items, and calculate the interests of users on each item category to construct the item category similarity of users. Then it combines the item category similarity with the user rating similarity to construct the user integrated similarity. Finally it looks for the nearest neighbors of users to get the best recommendation effect.

USER-BASED COLLABORATIVE FILTERING

Firstly, user-based collaborative filtering takes advantage of the user rating matrix to calculate the similarity between users. Then it looks for the nearest neighbors of target user and predicts the ratings. Finally, it generates recommendations to target user according to the predicted values. The algorithm is mainly divided into the following three steps:

1) Get the User Rating Matrix: Generally, we can get a $m \times n$ user rating matrix where *m* denotes the number of users and *n* denotes the number of items. The matrix element denotes the rating of user *u* on item *i*. The user rating matrix is shown in TABLE1.

TABLE 1 : User rating matrix

	$Item_1$	⋯	$Item_j$	⋯	$Item_n$
$User_1$	$R_{1,1}$	⋯	$R_{1,j}$	⋯	$R_{1,n}$
⋮			⋮		⋮
$User_i$	$R_{i,1}$	⋯	$R_{i,j}$	⋯	$R_{i,n}$
⋮			⋮		⋮
$User_m$	$R_{m,1}$	⋯	$R_{m,j}$	⋯	$R_{m,n}$

2) Calculate Similarity and *K* Nearest Neighbors: We can calculate the similarity between users according to user rating matrix and obtain *K* nearest neighbors for the target user by the similarity from big to small.

3) Generate Recommendations: After getting the nearest neighbors of target user, we can obtain the ratings of users on any item and the *Top-N* recommend set by prediction formula.

Similarity calculation plays a vital role in the entire recommendation process. Selecting the similarity calculation method appropriately can effectively improve the quality of the recommendation of the recommender systems.

A. User Similarity Calculation Method

Similarity calculation is the most crucial step in collaborative filtering recommendation algorithm. There are mainly three ways to calculate the similarity between users[9]: cosine-based similarity, adjusted-cosine similarity, and correlation-based similarity.

1) Cosine-based Similarity: Similarity between users can be measured by vectorial angle cosine. The higher the cosine value between two users is, the higher the similarity degree they have. The ratings of a user are regarded as a *n* dimensional space vector. If a user does not rate an item, the rating on the item is 0. Formally, the cosine similarity calculation formula is

$$sim(u,v) = cos(\boldsymbol{u},\boldsymbol{v}) = \frac{\boldsymbol{u}\cdot\boldsymbol{v}}{\| \boldsymbol{u} \|_2 \times \| \boldsymbol{v} \|_2} = \frac{\sum_{i\in I_{uv}} R_{ui}\cdot R_{vi}}{\sqrt{\sum_{i\in I_{uv}} R_{ui}^2}\sqrt{\sum_{i\in I_{uv}} R_{vi}^2}} \quad \textbf{(1)}$$

where $sim(u,v)$ denotes the similarity between user *u* and *v*. I_{uv} denotes the item set that user *u* and *v* both rated. Vector *u* and *v* respectively denote the ratings of user *u* and *v* on I_{uv}. R_{ui} and R_{vi} respectively denote the ratings of user *u* and *v* on item *i*.

2) Adjusted Cosine Similarity: As the cosine similarity measure method does not take the rating scale of different users into account, therefore, adjusted cosine similarity offsets this drawback by subtracting the corresponding user average from each co-rated pair. Formally, the adjusted cosine similarity formula is

$$sim(u,v) = \frac{\sum_{i\in I_{uv}}(R_{ui}-\overline{R_u})\cdot(R_{vi}-\overline{R_v})}{\sqrt{\sum_{i\in I_u}(R_{ui}-\overline{R_u})^2}\sqrt{\sum_{i\in I_v}(R_{vi}-\overline{R_v})^2}} \quad \textbf{(2)}$$

where I_{uv} and I_v respectively denote the item set that user *u* and *v* rated. I_{uv} denotes the item set that user *u* and *v* both rated. R_{ui} and R_{vi} respectively denote the ratings of user *u* and *v* on item *i*. $\overline{R_u}$ and $\overline{R_v}$ respectively denote the average ratings of user *u* and *v*.

On account of the extreme sparsity of user rating data, there are a lot of users who have few co-ratings with others. Therefore, (2) can not accurately calculate the rating similarity between users. For the disadvantages of (2), this paper introduces a common factor to correct (2) as

$$sim_\lambda(u,v) = \frac{|I_u \cap I_v|}{max(\lambda, |I_u \cap I_v|)} sim(u,v) \quad \textbf{(3)}$$

where $|I_u \cap I_v|$ denotes the number of item that user *u* and *v* both rated. λ is a parameter that we set. When the number of items user *u* and *v* both rated is more than λ, that is, the number of co-rated items is large, we still use the similarity calculation method of (2). Otherwise, we use the common factor $\frac{|I_u \cap I_v|}{\lambda}$ to modify the similarity calculation formula.

3) Correlation-based Similarity: Similarity between user *u* and *v* can be obtained by calculating Pearson correlation. Formally, the correlation-based similarity formula is

$$sim(u,v) = \frac{\sum_{i\in I_{uv}}(R_{ui}-\overline{R_u})\cdot(R_{vi}-\overline{R_v})}{\sqrt{\sum_{i\in I_{uv}}(R_{ui}-\overline{R_u})^2}\sqrt{\sum_{i\in I_{uv}}(R_{vi}-\overline{R_v})^2}} \quad \textbf{(4)}$$

where I_{uv} denotes the item set that user *u* and *v* both rated. R_{ui} and R_{vi} respectively denote the ratings of user *u* and *v* on item *i*. $\overline{R}_u$ and $\overline{R}_v$ respectively denote the average ratings of user *u* and *v*.

B. Generate Recommendations

After working out the nearest neighbors of target user by user similarity measure method, we can calculate two types of recommendation results.

1) The ratings of target user on any item: We set the nearest neighbors of target user *u* as N_u, thus, the prediction on item *i* of target user *u* is

$$P_{ui} = \overline{R}_u + \frac{\sum_{v\in N_u} sim(u,v)(R_{ui}-\overline{R}_v)}{\sum_{v\in N_u} sim(u,v)} \quad \textbf{(5)}$$

where $\overline{R}_u$ and $\overline{R}_v$ respectively denote the average ratings of user *u* and *v*.

2) *Top–N* recommendation set: After predicting ratings on different items of user *u*, we take the first *N* items which have the highest values as a *Top–N* recommendation set.

USER INTEGRATED SIMILARITY BASED COLLABORATIVE FILTERING

In the actual e-commerce system, commodities are generally divided into several different categories. Users often only browse or buy commodities in specific categories which they are interested in, and rate commodities which they concern. Therefore, we can consider that there are certain similarity between users who concern the common categories.

However, the traditional user-based collaborative filtering algorithm does not consider the relationship between user and item category, only relying on single user rating data to calculate the similarity between users, leading to the decline of recommendation quality. Assuming that the ratings on item set $I = \{i_1, i_2, i_3, i_4, i_5\}$ of user *u* and *v* are shown in TABLE 2.

TABLE 2 : Rating table of user u and v

	c_1	c_1	c_1	c_1	c_1
	i_1	i_2	i_3	i_4	i_5
u	4	4	4		
v				4	4

In this case, item i_1, i_2, i_3, i_4, i_5 belong to the category c_1; the rating based on 5-point; 1 is the lowest, and 5 is the highest; null value indicates no rating.

According to TABLE 2, we can obtain that user *u* rated item i_1, i_2, i_3 and user *v* only rated item i_4 , i_5. If we only calculate it by the single user rating data, the similarity of user *u* and *v* is 0. However, actually the items they rated belong to the same category c_1, and obviously there is a certain similarity between user *u* and *v*, that is they have a same interest. Therefore, the traditional similarity calculation method does not mine the intrinsic characteristics of items. Aiming at this problem, this paper proposes a collaborative filtering recommendation algorithm based on user integrated similarity.

A. Definition of Item Category Interestingness Eigenvector

Given user *u*, $N(u)$ denotes the item set which user *u* rated. Given category *c*, $N(c)$ denotes the item set which is in category *c*. Introducing the item category interestingness eigenvector T_{uc}, the definition is as follows:

$$T_{uc} = \frac{|N(u) \cap N(c)|}{|N(u)|} \quad \textbf{(6)}$$

In this case, eigenvector T_{uc} denotes that items in category *c* which user *u* rated takes how much the proportion of all items user *u* rated. T_{uc} can be interpreted as the users' interests in category *c*, that is the item category interestingness.

However, the eigenvector above doesn't consider the impact of user rating impact on user's interest. Assuming that the ratings on item set $I = \{i_1, i_2, i_3, i_4, i_5\}$ of user *u* and *v* are shown in TABLE 3.

TABLE 3 : Rating table of user u and v

	c_1	c_1	c_2	c_1	c_3
	i_1	i_2	i_3	i_4	i_5
u	5	4	2	3	1
v	2	3	3	1	5

In this case, item i_1, i_2, i_4 belong to category c_1; item i_3 belong to category c_2; item i_5 belong to category c_3; the rating based on 5-point; 1 is the lowest, and 5 is the highest.

By using (6), we can get $T_{uc_1} = \frac{|N(u) \cap N(c_1)|}{|N(u)|} = \frac{3}{5}$, $T_{vc_1} = \frac{|N(v) \cap N(c_1)|}{|N(v)|} = \frac{3}{5}$.

Thus, the interestingness on item category of these two users is same. In fact, it can be found that rating of user *u* on item category c_1 is generally high, but the rating of user *v* on category c_1 is generally low. Obviously, however, the interest of user *u* on item category c_1 is higher than user *v*'s. To solve this problem, the eigenvector above can be further adjusted to

$$T_{uc} = \frac{|N(u) \cap N(c)|}{|N(u)|} \cdot \frac{\overline{R}_{uc}}{r_{uc}^{max}} \quad \textbf{(7)}$$

where $\overline{R}_{uc}$ denotes the average rating of user *u* on items in category c. γ_{uc}^{max} denotes the upper limit value of user *u* on items in category *c*; $\frac{\overline{R}_u}{r_{uc}^{max}}$ can be expressed as rating level of user *u*. For instance, in TABLE 3, $\overline{R}_{uc_1}$ of user *u* is 4, and $\gamma_{uc_1}^{max}$ is 5.

Thus, it can be calculated by (7) as follows:

$$T_{uc_1} = \frac{|N(u) \cap N(c_1)|}{|N(u)|} \cdot \frac{\overline{R}_{uc_1}}{r_{uc_1}^{max}} = \frac{3}{5} \times \frac{4}{5} = 0.48,$$

$$T_{vc_1} = \frac{|N(v) \cap N(c_1)|}{|N(v)|} \cdot \frac{\overline{R}_{vc_1}}{r_{vc_1}^{max}} = \frac{3}{5} \times \frac{2}{3} = 0.4.$$

The obtained results are consistent with the actual, that is, compared to user *v*, user *u* has a higher interest in category c_1.

B. Definition of User's Item Category Interestingness Similarity

We can obtain the user's item category interest matrix by the item category interestingness eigenvector defined above as follows:

$$T = \begin{bmatrix} T_{1,1} & T_{1,2} & \cdots & T_{1,k} \\ T_{2,1} & T_{2,2} & \cdots & T_{2,k} \\ \vdots & \vdots & & \vdots \\ T_{n,1} & T_{n,2} & \cdots & T_{n,k} \end{bmatrix}$$

In this case, *n* denotes the number of users; *k* is the number of item category; $T_{n,k}$ denotes the interest of user u in category *k*. The interest of user *u* can be denoted with vector $T_u = (T_{u,1}, T_{u,2}, \cdots, T_{u,k})$. Therefore, for user *u* and *v*, by using (1), the category interestingness similarity between item *i* can be calculated as

$$sim_c(u,v) = \frac{\sum_{k=1}^{k} T_{u,k} \cdot T_{v_k}}{\sqrt{\sum_{k=1}^{k} T_{u,k}^2}\sqrt{\sum_{k=1}^{k} T_{v,k}^2}} \quad \textbf{(8)}$$

where $sim_c(u,v)$ denotes the item category interestingness similarity between user *u* and *v*; *k* denotes the number of categories of items; $T_{u,k}$ and $T_{v,k}$ respectively denote the interests in category *k* of user *u* and *v*.

C. Definition of Integrated Similarity

The traditional user-based collaborative filtering algorithm only considers the similarity between users from single user rating data, and can not reflect the influence of item category on similarity between users. The interestingness similarity from the perspective of item category proposed in this paper, can accurately mine the correlation between the inherent features of items and the interests of users, that is, two users who have higher item category interestingness similarity have similar interests. The algorithm proposed in this paper combines the user rating similarity with the item category interestingness similarity to get the integrated similarity. It is defined as

$$sim(u,v) = (1-\alpha)sim_c(u,v) + \alpha sim_p(u,v) \quad \textbf{(9)}$$

where $0 \leqslant \alpha \leqslant 1$. $sim_c(u,v)$ is the item category interestingness similarity calculated by (8). $sim_p(u,v)$ is the user rating similarity calculated by (3). The Value of α can reflect the each importance of item category interestingness similarity and user rating similarity to the user similarity. When α is 1, we get (3), which only calculates the rating similarity; when α is 0, we get (8), which only considers the interest similarity of users.

By using (9) we can obtain the integrated similarity matrix *S*, which can be expressed as

$$S = \begin{bmatrix} sim(1,1) & sim(1,2) & \cdots & sim(1,m) \\ sim(2,1) & sim(2,1) & \cdots & sim(2,m) \\ \vdots & \vdots & \ddots & \vdots \\ sim(m,1) & sim(m,1) & \cdots & sim(m,m) \end{bmatrix}$$

where *m* is the number of users, and the matrix element denotes the integrated similarity between two users.

D. Algorithm Description

Input: item set *I*, user rating data, the number *K* of user neighbor, the number *N* of element of recommendation set

Output: recommendation set *Top-N* of the target user *u*

The Algorithm Process:

Get the user rating matrix $R_{m\times n}$ by using the user rating data, in which *m* is the number of users, *n* is the number of items.
Divide item set *I* into *k* categories by using of the existing category system or clustering algorithm, then get the category set $C=\{c_1,c_2,\cdots,c_k\}$. *k* is the number of categories.
Get the item category interestingness matrix according to (6) and (7).
for (any user *u* and *v*) do
Get the user rating similarity $sim_p(u,v)$ according to (3);
Get the item category interestingness similarity $sim_c(u,v)$ according to (8);
Get the user integrated similarity $sim(u,v)$ according to (9).
for (each user *u*) do
According to the integrated similarity matrix *S*, find out the nearest neighbors set composed of first *K* users $N_u=\{u_1,u_2,\cdots,u_K\}, u\notin N_u$.
for (each user *u*) do
Calculate prediction ratings of *u* on non-rated items according to (5);
Sort the prediction ratings in ascending order;
Take the corresponding items of first *N* value to form a *Top-N* recommendation set.

EXPERIMENTAL EVALUATION

A. Data set

The experimental data set was provided by the MovieLens web site (http://movielens.umn.edu). The data set consists of 100,000 ratings from 943 users on 1682 movies, in which each user has rated at least 20 movies. The rating range is 1~5. The greater the rating is, the larger interest user have in movies. Namely, 1 represents the least favorite; 5 is the most favorite. The data set includes 19 (0 ~ 18) different types of movie categories. Each movie at least belongs to one category but can simultaneously belong to multiple categories. We only use type1 ~ 18 (Type 0 is abnormal record, which should be discarded).

The data set was divided into training set and test set in accordance with proportion of 80% and 20%. Based on this, we conducted a 5- fold cross experiment and take the average of five sets of data to verify. The sparse level of the entire data set is

$$1-\frac{100000}{943\times 1682}=0.9370.$$

B. Evaluation Metrics

Evaluation metrics of quality of recommender systems mainly include statistical accuracy metrics and decision support accuracy metrics. MAE [10] (Mean Absolute Error) in statistical accuracy metrics is the most common standard to measure the accuracy of recommendation in collaborative filtering algorithm. MAE is mainly used to calculate the absolute difference between ratings and predictions in test set. The smaller the value of MAE is, the higher the quality of the recommendation system will be.

The prediction set of user rating is expressed as$\{p_1,p_2,\cdots,p_N\}$, and the corresponding actual user rating set is expressed as $\{q_1,q_2,\cdots,q_N\}$. Formally, MAE is defined as

$$MAE=\frac{\sum_{i=1}^{n}|p_i-q_i|}{N} \quad \textbf{(10)}$$

C. Experimental Parameters

The common factor parameters in (3) is preset to adjust the calculation formula of user similarity, which can be used to modify the problem that we can't accurately calculate the similarity between users because of the sparsity of user ratings, and can make the similarity calculation more

reasonable. In this experiment, we vary the value of λ from 20 to 80 with a step value of 20. As you can see from Figure 1, when λ is 40, MAE is the minimum.

In order to determine the weight of α in (9), in this experiment, we vary the value of α from 0 to 1 with a step value of 0.1. As is shown in Figure 2, when α is 0.6, MAE is the minimum, which achieved the best combination of weight and the best recommendation accuracy.

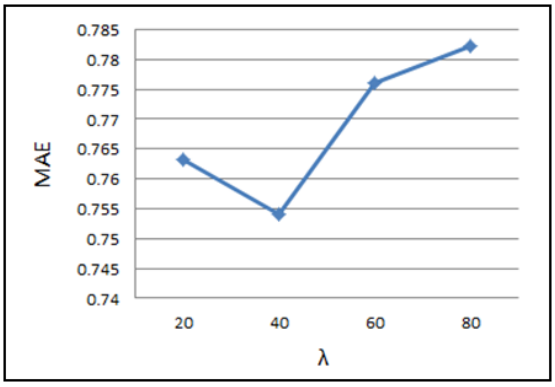

Figure 1 : Impact of λ

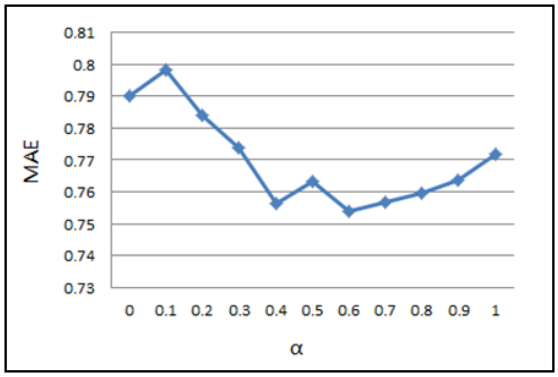

Figure 2 : Impact of α

D. Experimental Results

In order to verify the effectiveness of the algorithm proposed in this paper, by using the same data set, we compared the traditional user-based collaborative filtering algorithm (UBCF) with the user integrated similarity based collaborative filtering algorithm (UISBCF) proposed in this paper. The similarity calculation formula are all based on adjusted cosine similarity. The number of nearest neighbors are, in order, 5, 10, 20, 30, 40, 50. The experimental parameters are set according to the optimal values discussed in previous section. The experimental results is shown in Figure 3.

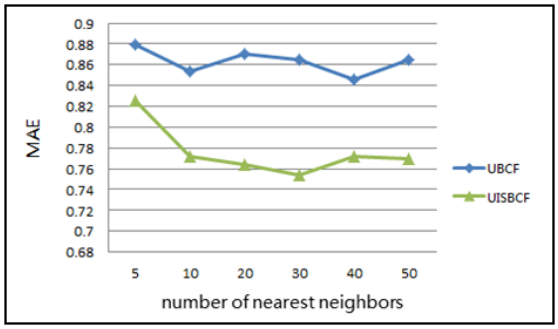

Figure 3 : Comparison of MAE of Recommendation Algorithms

It can be seen from Figure 3 that, with the increase of the number of nearest neighbors, the algorithm (UISBCF) proposed in this paper has been better than the user-based collaborative filtering algorithm (UBCF) and has a smaller MAE. Thus, we can infer that the algorithm proposed in this paper can effectively improve the quality of the recommendation of recommender systems and provide users with more accurate personalized recommendations.

CONCLUSIONS

In allusion to the inaccuracy of traditional similarity calculation method in the case of extreme sparsity of user rating data, this paper proposes a collaborative filtering recommendation algorithm based on user integrated similarity. The algorithm first modifies the similarity calculation formula by introducing the common factor, and on this basis it classify items by item category information. Then it constructs the item category interestingness eigenvector to get the user's item category interestingness similarity. Furthermore, it combines the user rating similarity to construct the integrated similarity. Finally, it looks for the nearest neighbors to make recommendations. The experimental result indicates that this algorithm can effectively improve the quality of the recommendation of recommender systems.

ACKNOWLEDGMENT

This study was supported in part by Natural Science Foundation Fundamental Research Project of Shaanxi Provence of China No.2012JM8037.

REFERENCES

[1] P.B.Kantor, F.Ricci, L.Rokach et al.; Recommender systems handbook[M]. Springer, **(2011)**.
[2] X.Su, T.M.Khoshgoftaar; A survey of collaborative filtering techniques[J]. Advances in artificial intelligence, **2009**, 4 **(2009)**.
[3] R.Pan, Y.Zhou, B.Cao et al.; One-class collaborative filtering[C]//Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. IEEE, 502-511 **(2008)**.
[4] G.Linden, B.Smith, YorkJ.Amazon; com recommendations: Item-to-item collaborative filtering[J]. Internet Computing, IEEE, **7(1)**, 76-80 **(2003)**.
[5] J.B.Schafer, D.Frankowski, J.Herlocker et al.; Collaborative filtering recommender systems[M]//The adaptive web. Springer Berlin Heidelberg, 291-324 **(2007)**.
[6] P.Massa, P.Avesani; Trust-aware collaborative filtering for recommender systems[M]//On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. Springer Berlin Heidelberg, 492-508 **(2004)**.
[7] B.Sarwar, G.Karypis, J.Konstan et al.; Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. ACM, 285-295 **(2001)**.
[8] G.Adomavicius, A.Tuzhilin; Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. Knowledge and Data Engineering, IEEE Transactions on, **17(6)**, 734-749 **(2005)**.
[9] J.S.Breese, D.Heckerman, C.Kadie; Empirical analysis of predictive algorithms for collaborative filtering[C]//Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 43-52 **(1998)**.
[10] F.Cacheda, V.Carneiro, D.Fernández et al.; Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems[J]. ACM Transactions on the Web (TWEB), **5(1)**, 2 **(2011)**.

[11] Tian-Shi Liu; Received his B.S. degree from Northwest University, China in 1982, and obtained his M.E. degree from Xi'an Jiaotong University, China in 1985, and obtained Ph.D. degrees from Northwestern Polytechnical University University, China in 2005. Now he is a Professor and Master Degree supervisor in School of Computer Science at Xi'an Shiyou University, Xi'an, China. His current research interests include distributed systems, information system, computer networks, and communication optimization algorithm.

[12] Nan-Jun Sun; Received his B.E in software engineering from Xi'an Shiyou University, Xi'an, Shaanxi, China, in 2012. Currently, he is a master's candidate in computer technology at Xi'an Shiyou University, Xi'an, Shaanxi, China. His research interests include management information system, data mining and recommender systems.

[13] Liu-Mei Zhang; Received his B.E in computer science from Air-force Engineering University, Xi'an, Shaanxi, China, in 2003, and obtained his M.I.T degree in database management from The School of Information Technology, University of Sydney, Sydney, Australia, in 2007. Currently, he is a PhD candidate in computer architechture at Xidian University, Xi'an, Shaanxi, China. His research interests include data mining, wireless sensor networks, and intelligent computing.