International Journal of u- and e- Service, Science and Technology
Vol.9, No. 2 (2016), pp.185-194
http://dx.doi.org/10.14257/ijunesst.2016.9.2.19

Research on User-item Rating based on Collaborative Filtering Algorithm

Song Li

Institute of Engineering, Mudanjiang Normal University
Mudanjiang 157011, China
songli1111@163.com

Abstract

Traditional collaborative filtering methods just use user-item rating matrix to generate recommendations, and lead to difficult to computer the similarity because of the data sparsity. We propose a hybrid collaborative filtering algorithm combining the rating matrix and item attributes. First, we design a user similarity measurement method by computing the user's preference to different item attributes, this approach is consistent with the true relationship between users, and also can effectively alleviate the issue of rating matrix sparse. Then, when computing the similarity of two users, we combine the Pearson correlation and the items attribute preference similarity, with a weighting coefficient "w" to balance the importance of two parts. Experiments show that this algorithm effectively solves the problem of data sparsity, and outperforms better when the sparsity is more serious, compared to the traditional CF algorithms.

***Keywords**: Recommendation Systems, Collaborative Filtering, User-item rating matrix*

1. Introduction

Collaborative filtering algorithm generates recommendations by depending on user's comments on resource items. If users don't appraise too much about one item, it's not possible to produce accurate recommendations only based on very few evaluations. What worse is if users with similar preference don't rate one item jointly, traditional collaborative filtering algorithms will not recommend precisely. This is the sparseness problem of evaluation data encountered by the collaborative filtering system. In practical recommendation system, some big e-commerce websites like Amazon.com and Netflix.com contain mega or millions of resource items. Items rated by one active user cannot exceed one percent of the total amount [1]. When rating matrix is too sparse, it's very difficult to find out similar user group by only relying on user-item rating data, downgrading the recommendation effect.

To alleviate the impact of data sparsity, researchers developed many solutions. The commonest one is missing value filling of unrated data. The simplest way is to use one specific missing value (which is generally made as rating mean value of the item or user) to replace null values in the rating matrix. Experiments show that the method improves in certain degree the recommendation precision of the system. But a big shortcoming is one user can't hold the same attitude to all items. The mean value filling ignores different users' interest and hobbies. It can't solve data sparseness problem from the root.

2. Problem and Existing Solutions

Some researchers incorporate user/item information into collaborative filtering system to eliminate data sparsity based on the suggestion of contents. Content-based recommendation is renamed recommendation based on semantics. It's independent of user's comments on items; instead, it calculates similarity between items by referring to item attribute, status and relationship, as thus to generate recommendations. Assuming $i_p = \{p_1, p_{2,} ... p_m\}$ and $i_q = \{q_1, q_2, ..., q_m\}$ represent the attribute vector of two items, then used content-based recommendation algorithm. The similarity of project i_p and i_q is calculated:

$$sim(i_p, i_q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} sim(i_p, i_q)}{|i_p| \times |i_q|} \quad (1)$$

Content-based similarity relies merely on item attribute to make recommendations, which will easily lead to excessive specialization. To be specific, in a book recommendation system, when it knows one user's special love of science fictions, the system will constantly recommend such books to the user based on book attribute similarity. But on this regard, user acquired recommendations will be restricted to involve one type of commodity. Comparatively, collaborative filtering can find user's potential interests. Content-based recommendation and the one based on collaborative filtering are usually used together [2-4].

The paper [5] stated one collaborative filtering algorithm combing item-based and user-based to reduce matrix sparseness, to find the similar user set u_a and item set i_k for the missing rated item r_{ui}, which are respectively defined as:

$$S(u) = \{u_a \mid sim'(u_a, u) > \eta, u_a \neq u\}$$
$$S(i) = \{i_{ka} \mid sim'(i_k, i) > \theta, i_k \neq i\} \quad (2)$$

where, η and θ stand for the threshold of similarity between user and item. Only if user's similar degree is bigger than η or item's similar degree is over θ, they can be added to user or item's neighboring collection. The final recommendation result of the algorithm is obtained based on the weighting of item-based prediction result and user-based prediction result. The result reflects that the method performs better than conventional collaborative filtering algorithms. However, it requires adjusting many parameters and doing one by one in practice.

In addition to many model based collaborative filtering method is used to solve the problem of sparse matrix, the Bayesian network [6-7], clustering [8-10] and neural network [11], association rules and methods, for example, also made a good recommendation effect.

Traditional collaborative filtering techniques consider only use-item rating matrix data, easily susceptible to data sparseness and causing inaccurate recommendation results. So here on the basis of user-item rating matrix, we introduce item attribute matrix. Further on, we propose one mixed collaborative filtering algorithm based on user rating similarity and item attribute preference similarity to overcome data sparseness. Its basic idea is: the similarity among users is related to not only user's scoring of item and also user's preference degree for some sort of time. When two users' rated item attributes are alike, it's believed they have higher similarity. By considering overall the similarity of user rating and item attribute preference similarity, it avoids the shortage of traditional similarity computing methods calculating by only depending on user rating, abating negative impacts brought by rating data sparseness.

3. Hybrid Collaborative Filtering based on user Rating and Item Attribute Preference

3.1. Measure of User Similarity based on Item Attribute Preference

In reality, user similarity is not only associated with user's evaluation of items, but also connected with its preference for some item. When items assessed by two users have similar attribute, it's thought that they share higher similarity. One item may have several attributes, e.g. one movie may be affection film or comedy. Target user's attribute love of one item should have certain similarity with its neighbors' attribute preference for the item. For instance, if there're more comedies of target user rated movies, it's more likely for the generated neighboring users to evaluate comedies. So the combined advantage of item attribute similarity and traditional calculating methods based on item can improve the recommendation precision of the system.

Suppose all item attributes can be expressed with one collection $\{Attr_1, attr_2, ...Attr_k\}$; features of each item can be described by one or more attributes in the collection. For n items, it's possible to create the item feature matrix A as seen in Table 1. A is a binary matrix; where $A(i, j) = 1$ represents item i has attribute j; $A(i, j) = 0$ means item i doesn't have attribute j.

Table 1. Item Attribute Table

Items	Attr1	Attr2	Attr3	…	Attrk
Item1	0	1	0	…	1
Item2	1	0	0	…	0
Item3	0	1	1	…	0
…	…	…	…	…	…
Itemn	1	0	0		1

The preference degree $I_{u,i}$ of user u for item attribute i, it can be expressed by formula 3:

$$I_{u,i} = score_{u,i} / score_u \tag{3}$$

Where, $score_{u,i}$ represents the sum of the score values for user u for the class item i, and $score_u$ represents the total score of the user u on all items. Assuming that the total of the item has k categories, the degree of preference of the user is calculated by formula 3, and the vector $I_u = (I_{u,1}, I_{u2,} I_{u}3, ..., I_{u,k})$ is obtained. The similarity of item attribute preference $sim_s(u,v)$ of user u and v can be calculated by formula 4:

$$sim_s(u,v) = \frac{\sum_{i=1}^{k} I_{u,i} I_{v,i}}{\sqrt{\sum_{i=1}^{k} I_{u,i}^2}\sqrt{\sum_{i=1}^{k} I_{v,i}^2}} \tag{4}$$

Table2 is the score list of one user commenting movies. To describe simply the problem, we assume there're two kinds of movies: action movie (Action1-Action4) and comedy (Comedy1-Comedy4); each movie can belong to only one kind. From the following table we know that user1 and user2 don't have commonly rated item. On this case, we can't calculate their similarity with traditional relevant coefficients. But in terms of marks, user1 and user2 think highly of action movies and poorly of comedies. Actually they have strong similarity. With the proposed similarity calculating method based on item attribute preference, we can get the result by equation 4: $sim_s(user1, user2) = 0.996$, well matching with their real relationship. Therefore, we think the similarity measuring approach based on item attribute preference can partially eliminate matrix sparseness.

Table 2. Ratings for Different Types of Videos

User	Action1	Action2	Action3	Action4	Comedy1	Comedy2	Comedy3	Comedy4
User1	5			5	2		1	
User2		5	5			2		2
User3	3		3		5			5

3.2. Workflow of the Algorithm

We proposed one hybrid collaborative filtering algorithm which bases on rating similarity and item attribute preference similarity. The technique includes these four steps:

Step 1 Calculate user-item rating matrix R and user-item attribute preference matrix S according to user rating table and item attribute table;

where, matrix R size is $m \times n$; m is the quantity of user; n is the quantity of item; r_{ij} means user i's mark on item j; if user doesn't grade, r_{ij}=0; matrix S size is $m \times k$; m is the number of user; k is the number of item attribute; $score_{i,j}$ refers to the score summation of the jth class of items rated by user i;

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}, \quad S = \begin{bmatrix} Score_{11} & Score_{12} & \cdots & Score_{1k} \\ Score_{21} & Score_{22} & \cdots & Score_{2k} \\ \vdots & \vdots & & \vdots \\ Score_{m1} & Score_{m2} & \cdots & Score_{mk} \end{bmatrix} \qquad (5)$$

Step 2 Calculate user's preference vector of item attribute;

Use matrix S to compute each user's item attribute preference vector $I_u = (I_{u,1}, I_{u2,} I_{u3}, ..., I_{u,k})$ with equation 3; then with equation 4, get user u and v's preference similarity $sim_s(u,v)$ of item attribute;

Step 3 With reference to Pearson similarity and that based on item attribute preference, measure comprehensively user's similarity.

Since Pearson correlation is the mostly used similarity measuring method of collaborative filtering algorithms. In calculating user rating data similarity, we use Pearson correlation to do that.

For matrix R, we utilize Pearson correlated equation 2 to compute the similarity $sim_R(u,v)$ between user u and v; then based on $sim_R(u,v)$ and $sim_s(u,v)$ together with one weighting w, control the importance degree; the equation for calculating the similarity between the final user u and v is shown as follows:

$$sim(u,v) = w \times Sim_R(u,v) + (1-w) \times sim_s(u,v)$$

$$= w \times \frac{\sum_{i\in C}(r_{u,i}-\overline{r}_u)(r_{v,i}-\overline{r}_v)}{\sqrt{\sum_{i\in C}(r_{u,i}-\overline{r}_u)^2\sum_{i\in C}(r_{v,i}-\overline{r}_v)^2}} + (1-w)\times\frac{\sum_{i=1}^{k} I_{u,i}I_{v,i}}{\sqrt{\sum_{i=1}^{k} I_{u,i}^2}\sqrt{\sum_{i=1}^{k} I_{v,i}^2}} \qquad (6)$$

Step 4 Generate recommendations

For one target user u, choose from user similarity matrix Sim the k most similar users to form k neighboring collection KNB of it; user u's predicted rating $P_{u,i}$ of item i is arrived through the equation:

$$P_{u,i} = \overline{R}_u + \frac{\sum_{v\in KNB} Sim(u,v)\times(R_{v,i}-\overline{R}_v)}{\sum_{v\in KNB}(Sim(u,v))} \qquad (7)$$

4. Experiment Design and Analysis

4.1Experimental Data

We used movie rating dataset provided by MovieLens. In the dataset, researchers classified movie attribute to 18 types. Action, Adventure, Animation, Childrens, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western. Each movie has one or more attributes [12].

The experiment chose 100000 rating data by 943 users for 1682 films as experimental dataset. Each user commented at least 20 movies. The dataset's sparse level is:

$$1-100000/(943\times1682)=0.9365$$

Apparently the dataset is very sparse. Choose randomly 80% of it as training data and the rest 20% as testing data. The model performance is evaluated still with the commonest mean absolute error (MAE), which is obtained by the equation:

$$MAE=\left(\sum_{I=1}^{N}|\,p_i-q_i\,|\right)/N$$

.

Where, $\{p_1,p_2,...,p_N\}$ is predictive score, $\{q_1,q_2,...,q_N\}$ is actual score. The smaller the MAE, the higher the quality of the recommendation..

4.2 Experimental Results and Discussion

In Formula 6, w is a weight for adjusting user's Pearson correlation and item attribute preference similarity; w's value affects a lot the recommendation precision of the system. So it requires repetitive tests as to modify its value. Since w is chosen from the range [0,1], in the experiment here, we make w from [0,1]. Each time we increase it by 0.1, we can see its influence on the recommendation result of the system. Experimental results are shown in Figure 1:

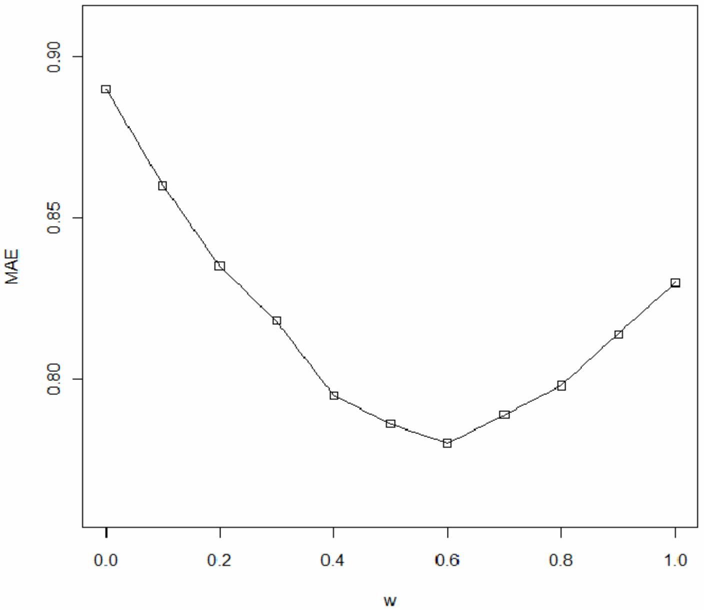


Figure 1. The Influence of Weight W on the Recommendation Accuracy

From Fig. 2, when w=0.6, the recommendation error MAE is the lowest with testing data, with best recommendation accuracy. So in the following comparative tests, we set weight w=0.6.

Next, we'll validate the effectiveness of the proposed hybrid collaborative filtering algorithm and compare it with traditional method based on Pearson correlation and cosine similarity collaborative filtering, in terms of recommendation quality MAE. The number of neighbors is made from 4 to 60 at interval of 4; weight w is 0.6. Experimental results are shown in Figure 2:

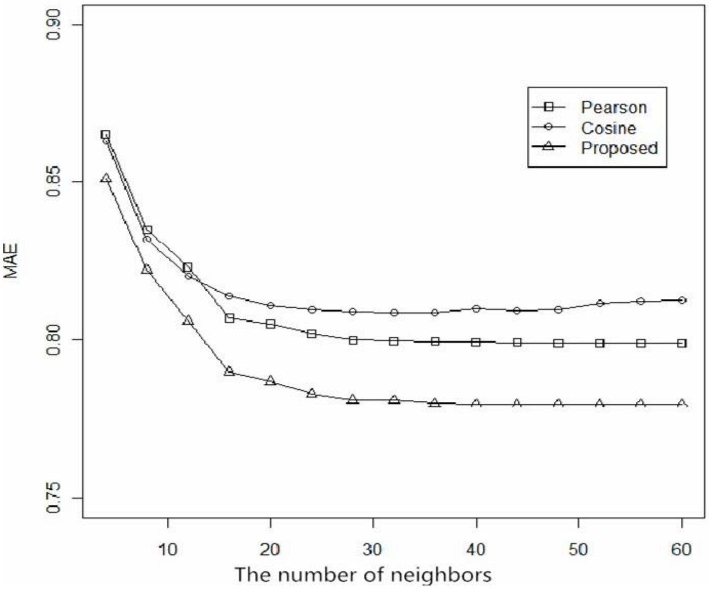


Figure 2. The Comparison of the Accuracy of the Proposed Method and the Traditional Recommendation Method

As seen from Fig. 2, for different number of neighbors, the proposed method reaches the least MAE value and its recommendation accuracy improves remarkably than the other two.

Moreover, with regards to different data sparsity, we examined performance of the proposed method and traditional ones. We took sparse sampling of training set and testing set, with the density respectively 80%, 60% and 40%, representing the identical percentage of sampling data density of the original data. The experimental results are compared with that shown in Figure 3 to figure 5:

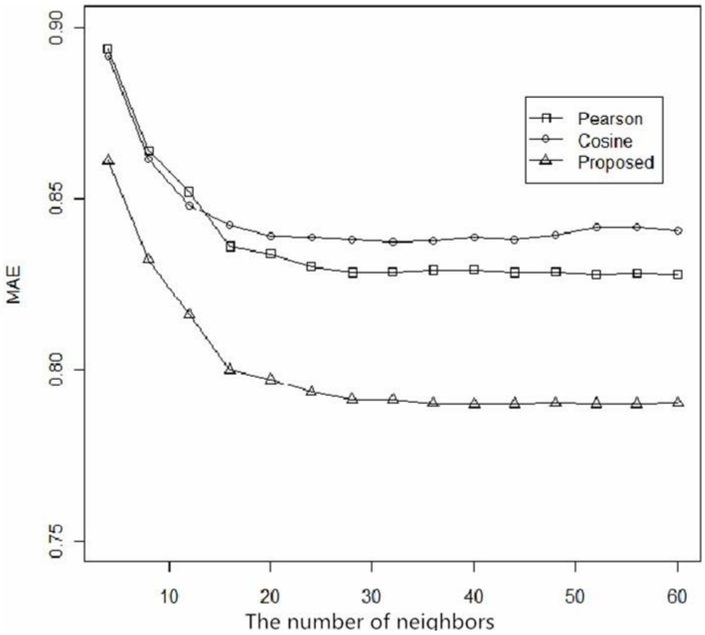


Figure 3. MAE Contrast of 80% Sparse Sampling

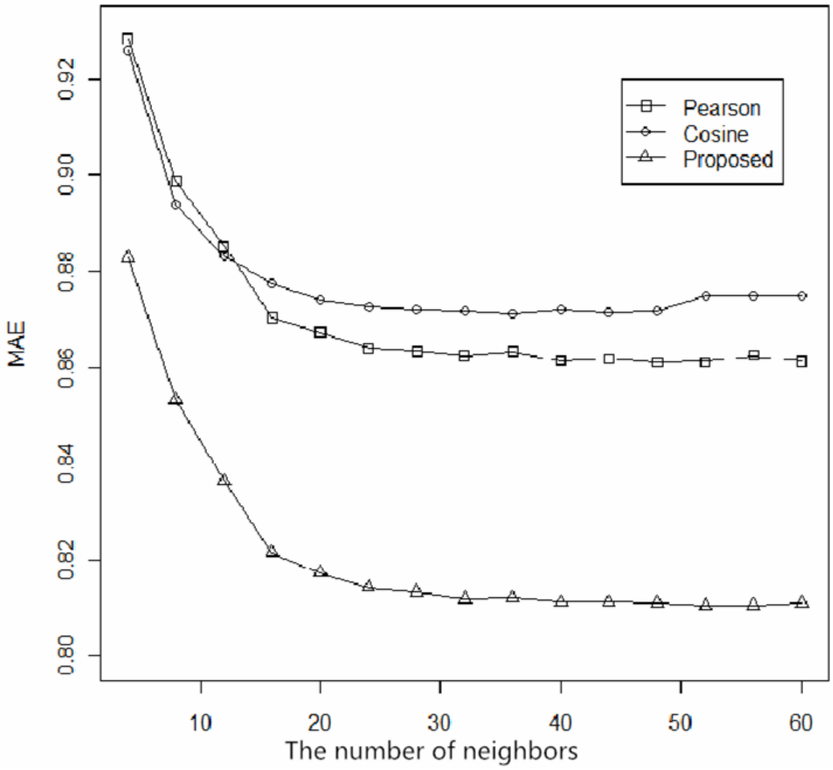


Figure 4. MAE Contrast of 60% Sparse Sampling

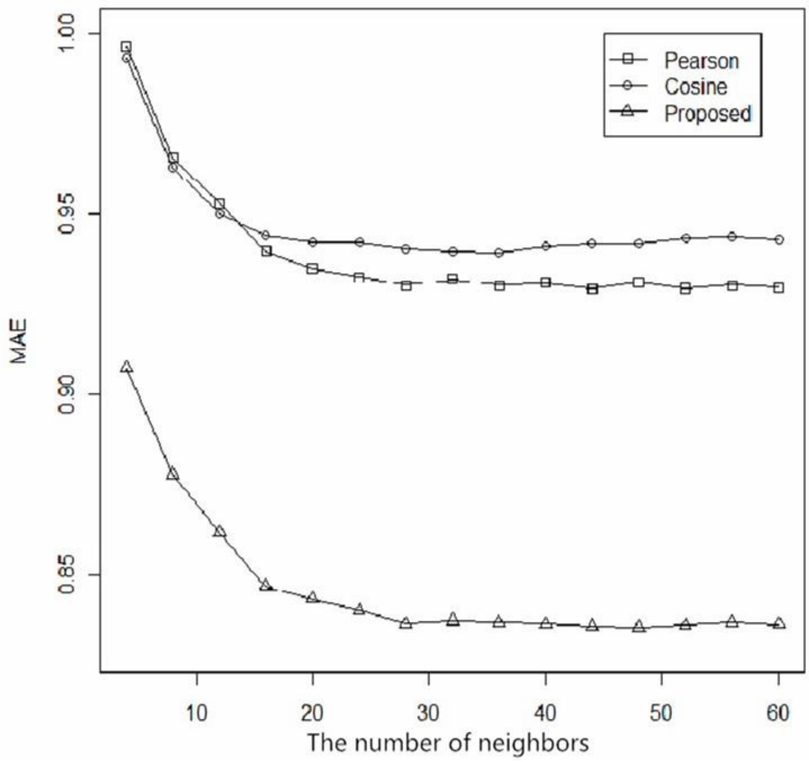


Figure 5. MAE Contrast of 40% Sparse Sampling

From Fig.3 to Fig.5, we find firstly, for different data sparsity, the proposed collaborative algorithm performed better than all traditional peers; next, with data becoming sparser, the method tended to improve gradually; when data become sparse as high as 40%, it improved the most obviously. It implies that by introducing item attribute information, the approach overcomes well shortcomings faced by traditional methods which calculate similarity by relying only on user rating. Even though two users don't common the same item, the technique can get their similarity as per item attribute, diminishing negative impacts caused by rating data sparsity and enhancing recommendation precision.

5. Conclusion

Traditional collaborative filtering techniques consider only user-item rating matrix data. They are easily affected by data sparseness and thus the recommendation result is not accurate. To address it, we introduced item attribute matrix on the basis of user-item rating matrix and presented one hybrid collaborative filtering algorithm based on user rating similarity and item attribute preference similarity, which alleviates data sparseness. First of all, with item attribute information, we designed one user similarity measuring method based on item attribute preference; it's consistent with the real relationship between users; also it can ease effectively user rating data sparseness; then in measuring user similarity, it considers overall user rating similarity and similarity of user's preference for item attribute; it controls their importance degree with one weight w. Through experiments, for different sparse data, the method here realized better recommendation accuracy than traditional ones. What's pleasing is as data become sparser, its performance improves more remarkably. The extraction of item attribute features is a field-related question. In the future, we'll discuss about how to fetch automatically and describe properly such features.

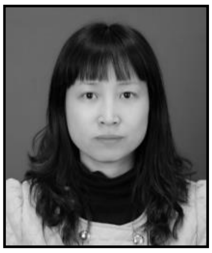
Acknowledgement

This work was supported by Mudanjiang Normal College science and technology research projects. No. QY2014001

References

[1] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms", Proc. Of the WWW Conference, **(2001)**.
[2] H. Song Hantao, L. Cui and Y. Lu, based on semantic similarity of resource collaborative filtering technology research, Journal of Beijing Institute of Technology, vol. 5, **(2012)**, pp. 45-49
[3] K. Tso and L. S. Thieme, "Attribute-aware collaborative filtering", From Data and Information Analysis to Knowledge Engineering, **(2006)**, pp. 614-621
[4] B. M. Kim, Q. Li and C. S. Park, "A new approach for combining content-based and collaborative fiters, Journal of Intelligent Information Systems, vol. 27, no.1, **(2006)**, pp. 79-91
[5] H. Ma, I. King and M. R. Lyu, "Effective missing data prediction for collaborative filtering", Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, New York: ACM, **(2007)**.
[6] J. S. Breese, D. Heckerman and C. Kadie, "Empirical analysis of Predictive Algorithms for collaborative filtering", Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, **(1998)**.
[7] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple Bayesian classifier", Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence, **(2000)**.
[8] A. Kohrs and B. Merialdo, "Clustering for collaborative Filtering Applications", Proceedings of CIMCA, Vienna: IOS Press, **(1999)**.
[9] G. Xue, C. Lin and Q. Yang, "Scalable Collaborative Filtering Using Cluster-based Smoothing", Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval,Brazil:ACM Press, **(2005)**.
[10] H. Wang, J. Gao and T. Z. Wang, "Personalized service, collaborative filtering recommendation based on user clustering", The application of computer, vol. 27, no. 5, **(2013)**, pp. 1225-1227
[11] Zhangfeng, "Using BP neural network to alleviate sparsity issue in collaborative filtering recommendation algorithm", Journal of computer research and development, vol. 43, no. 4, **(2006)**, pp. 667-672.
[12] MovieLens Dataset. http://MovieLens.umn.edu/.

Author



Song Li, She is an associate professor at Institute of Engineering of Mudanjiang Normal University. She is in the research of computer application.