International Journal of Grid and Distributed Computing
Vol. 9, No. 7 (2016), pp.23-32
http://dx.doi.org/10.14257/ijgdc.2016.9.7.03

Research on Collaborative Filtering Algorithm based on Cloud Computing

Dan Zhang

Institute of Technology, Mudanjiang normal university, Mudanjiang 157000, china,
zhangdanwyc@163.com

Abstract

In order to solve this problem of cloud model, this paper presents another new collaborative filtering recommendation algorithm by combining the item classification and cloud model. Firstly the algorithm utilizes the item classification information and cloud model to compute items inner-similarity, and then gets the scores from neighbor items which have the highest similarity and uses their scores to forecast the unrated inner-class items. Secondly, the neighbors of user are obtained by computing the inner-class user similarities in the cloud model, providing the final forecast grade and carrying out the recommendation.

Keywords*: Personalized Recommendation, Collaborative filtering, Item classification, Cloud computing.*

1. Introduction

The similarity measuring method based on cloud model overcomes the shortcomings of strict matching object attributes with traditional similarity measurement approaches [1-3] and avoids to a certain degree the negative impacts imposed by data sparsity. However, it does not consider that item class information has influences on the accuracy of obtained user interest [4-6], and not provide different interest recommendation sets subject to diversified user interests, causing the recommendation system not to have fundamental quality improvement [7-9]. To solve that problem, we propose the collaborative filtering improvement algorithm based on cloud model.

2. Collaborative Filtering Improvement Algorithm based on Cloud Model

2.1. Similarity Measurement Methods based on Cloud Model

E-commerce system acquires user's satisfaction degree of items with reference to user evaluation [10-11]. Suppose user rating is expressed by the discrete numerical value {1, 2, 3, 4, 5}, whose quantitative concept includes five levels as {very unsatisfied, unsatisfied, fair, satisfied, very satisfied}.

Definition 1. Item scoring frequency vector: counting the frequency of several users to a user that corresponds to the rating level, that is $I = \{I_1, I_2, I_3, I_4, I_5\}$, called the item scoring frequency vector of the target item; $I_1 - I_5$ refers to evaluation times of five levels by user for the item; based on item scoring frequency vector, with backward cloud algorithm, the qualitative knowledge converted from those scores can be obtained as to get item scoring feature vector. That is $C_I = (Ex, En, He)$.

ISSN: 2005-4262 IJGDC

Definition 2. User scoring frequency vector: calculating the frequency of one target user to all rated items which correspond to the rating level; That is $U=\{u_1,u_2,u_3,u_4,u_5\}$, u_1-u_5 stands for respectively rating times of five levels by user for relative item. It's called user scoring frequency vector.

In accordance to user scoring frequency vector, with backward cloud algorithm, the qualitative knowledge converted from those scores can be obtained and thus user scoring feature vector is acquired. That is $C_U=(Ex,En,He)$.

In the scoring feature vector of Definition 1-2, expected Ex means user's average satisfaction at item, which is preference level; the entropy En represents the concentration degree of scoring, which is the dispersion of scoring; hyper-entropy He refers to the stability degree of entropy.

Definition 3. Similarity of cloud: given clouds i and j 's numerical characteristics form respectively vector $\vec{V}_i$ and $\vec{V}_j$, between which the angle of their cosine is called similarity between i and j.

$$sim(i,j)=\cos(\overrightarrow{V_i},\overrightarrow{V_j})=\frac{\overrightarrow{V_i}\cdot\overrightarrow{V_j}}{\left\|\overrightarrow{V_i}\right\|\times\left\|\overrightarrow{V_j}\right\|} \tag{1}$$

Definition 4. Item similarity. The angle between the two items of the score feature vector called the term of their inter item similarity. Item i and item j similarity formula is:

$$sim(i,j)=\frac{C_{Ii}\cdot C_{Ij}}{\left\|C_{Ii}\right\|\times\left\|C_{Ij}\right\|}=\frac{Ex_i\cdot Ex_j+En_i\cdot En_j+He_i\cdot He_j}{\sqrt{Ex_i^{\,2}+En_i^{\,2}+He_i^{\,2}}\times\sqrt{Ex_j^{\,2}+En_j^{\,2}+He_j^{\,2}}} \tag{2}$$

.2. Score Forecasting Algorithm of the Cloud Model

In real e-business websites, commodities are generally divided into plentiful categories, *i.e.* item class matrix $C_{I,C}$, where I means commodity item; C means category. Any commodity belongs at least to one category or even several ones. Suppose item class $C=C_1\bigcup C_2\bigcup...\bigcup C_j$ is the collection of all item categories. Then item category matrix can be expressed like the following:

$$C_{I,C}=\begin{bmatrix} I_1 & C_{1,1} & \cdots & C_{1,j} \\ I_2 & C_{2,1} & \cdots & C_{2,j} \\ \cdots & \cdots & \cdots & \cdots \\ I_i & C_{i,1} & \cdots & C_{i,j} \end{bmatrix}$$

where I_1-I_i refers to the contained item, *i.e.* IID (identification); $C_{i,j}$ means if the item belongs to relative category, which is given 0 or 1; when $C_{i,j}$=1, indicating item i belongs to item category j; otherwise, not.

User-item rating matrix $R_{m\times n}$ is an m x n matrix; m rows suggest totally m users; n columns mean totally n items; matrix row-column crossing element $R_{UID,IID}$ stands for user UID's marks on item IID. Here in the experiment we use integer between 1~5, 1 for dislike, 5 for favorite.

Now we discuss $R_{UID,IID}$ in the matrix: when $R_{UID,IID}$ is not null, that is an integer between 1~5, $R_{UID,IID}$ is the rating $r_{UID,IID}$ made by UID for IID; when $R_{UID,IID}$ is null, *i.e.* no scores; then make inter-class rating prediction based on the category of $R_{UID,IID}$, in the following algorithm:

Algorithm1 Rating prediction algorithm of the cloud model
Input: user rating record Data and item information file Item
Output: prediction scoring $P_{UID,IID}$ by target user UID for IID

① Data pre-processing

Pre-treat user rating record Data to get user-item rating matrix $R_{m\times n}$; convert item information file Item to item class matrix $C_{I,C}$;

② Obtainment of class user-item rating matrix

With user rating list $R_{m\times n}$ and item class matrix $C_{I,C}$, we can get k class user-item rating matrix; behavior user UID listed as IID (which belongs to the class); matrix element is user UID's scoring value of IID.

③ Acquisition of inter-class item rating frequency vector

With the use of class user-item rating matrix got in the above step, reckon totally within each class the times $I_1 - I_5$ of scoring 1, 2, 3, 4, 5 points by all users for each item; and make as rating frequency vector I of inter-class item. $I = \{I_1, I_2, I_3, I_4, I_5\}$

④ Calculation of inter-class item rating feature vector

By means of reverse cloud algorithm, convert the inter-class item rating frequency vector $I = \{I_1, I_2, I_3, I_4, I_5\}$ reached in step 3 to the inter-class item rating feature vector $C_{Ii} = (Ex, En, He)$.

⑤ Computation of inter-class item similarity

For all classes, use equation (2) to calculate the similarity $sim(i, j)$ of inter-class item i and j, to get k class item similarity matrix $Sim[I]$:

$$Sim[\mathrm{I}] = \begin{bmatrix} sim(1,1) & sim(1,2) & \cdots & sim(1,m) \\ sim(2,1) & sim(2,2) & \cdots & sim(2,m) \\ \cdots & \cdots & \cdots & \cdots \\ sim(m,1) & sim(m,2) & \cdots & sim(m,m) \end{bmatrix}$$

where $Sim[I]$ is symmetric matrix, meaning the similarity of all users in the Ith class; row and column values are both user ID; row and column crossing value $sim(i, j)$ refers to the similarity degree between user i and j. Here we use dataset of totally 19 classes; removing class 0 (*i.e.* unknown abnormal data), we finally get 18 class similarity matrices.

⑥ Calculation of user UID's rating $P_{UID,IID}$ for item IID

Firstly, based on inter-class item similarity matrix, we make the first N items with the highest similarity as inter-class neighbor collection $M_{IID} = \{I_1, I_2, ..., I_N\}$ of IID, and $IID \notin M_{IID}$, the similarity value between item $I_1, I_2, ..., I_N$ and item IID becomes gradually small.

Then based on inter-class neighbor collection M_{IID}, we use the method [12] to predict user UID's marks of item IID:

$$P_{UID,IID} = \frac{\sum_{n \in M_{IID}} sim_{IID,n} * R_{UID,n}}{\sum_{n \in M_{IID}} \left| (sim_{IID,n}) \right|} \tag{3}$$

In it, M_{IID} is inter-class similarity item neighbor collection of item IID; $sim_{IID,n}$ is the similarity between item IID and inter-class item n; $R_{UID,n}$ is user UID's scoring on inter-class item n.

2.3. Collaborative Filtering Improved Algorithm based on Cloud Model

To have more accurate user neighbors and recommendation results, we propose the collaborative filtering improved algorithm based on cloud model with reference to the similarity measuring method of cloud model.

Algorithm2 Collaborative filtering improved algorithm based on cloud model

Input: user rating table $R_{m \times n}$ and item class matrix $C_{I,C}$

Output: target user UID's prediction score $P_{UID,IID}$ of IID

① Get inter-class user-item matrix

Based on user rating table $R_{m \times n}$ and item class matrix $C_{I,C}$, compute user item matrix $R_{m \times n}$, where m rows represent the number of users; n columns refer to the quantity of items; user UID's rating $R_{UID,IID}$ for item IID can be arrived at by the algorithm 1:

② Calculate inter-class user rating frequency vector

As per the inter-class user-item matrix got in step 1, sum up user's rating frequency vector $U = \{u_1, u_2, u_3, u_4, u_5\}$ in each category and regard as the inter-class user number i □; then by the inverse cloud algorithm, calculate rating feature vector $C_{Ui} = (Ex_i, En_i, He_i)$ of each user within the class.

③ Calculate inter-class user similarity matrix

According to category classification, use expression (3) to compute respectively the similarity $sim(C_i, UID)$ of inter-class user C_i and user UID, and have the user similarity matrix $Sim[i]$ of each category.

$$Sim[i] = \begin{bmatrix} sim(1,1) & sim(1,2) & \cdots & sim(1,m) \\ sim(2,1) & sim(2,2) & \cdots & sim(2,m) \\ \cdots & \cdots & \cdots & \cdots \\ sim(m,1) & sim(m,2) & \cdots & sim(m,m) \end{bmatrix}$$

④ Generate recommendation values

Use many items with the highest similarity in one class where target item IID belongs as the neighbor user of target user UID, that is, to look for user UID's nearest collection $C_{UID} = \{C_1, C_2, ..., C_k\}$, within the class, where user C_1 has the highest similarity with user UID, then C_2 and so on.

After the weighted mean value of each user's rating for item IID in the nearest neighbor collection C_{UID}, $P_{UID,IID}$ is concluded in the following way:

$$P_{UID,\text{IID}} = \overline{R_{UID}} + \frac{\sum_{u \in C_{UID}} sim(UID,u) \times (R_{u,IID} - \overline{R_u})}{\sum_{u \in C_{UID}} \left| sim(UID,u) \right|} \tag{4}$$

⑤ Make recommendation

When target item IID belongs to several categories, compute the recommendation value of each class by equation (4); then, take the average of all classes' recommendation values; next, get the round-off and use as the final recommendation value $P_{UID,IID}$.

3. Experiment Design and Discussion

3.1 Experimental Dataset

The dataset for this experiment is built about film sites by research team GroupLens in University of Minnesota, USA for studies on personalized recommendation, which is 100k open dataset provided by the site MovieLens. This set includes 100,000 rating records about 1682 movies by 943 users, of which each user comments at least 20 movies. According to the recent statistics, MovieLens has 40,000 registered users because of truthful and accurate data and abundant contents and at least 3,500 movies were evaluated. MovieLens dataset is widely applied for studies on various algorithms of personalized recommendation system. It is authoritative data source in the field [13].

The 100k MovieLens data set consists of the following files, file name and file contents as follows:

(1) u.data: user id | item id | rating | timestamp

The paper mainly includes the user serial number UID, the project serial number IID, the user's score for the project Rating, the timestamp of four. In the experiment, data preprocessing is obtained by the user - item score matrix $R_{m\times n}$, which is composed of $R_{m\times n}$, , $R_{m\times n}$ is is a matrix of m row n column, and the format is as follows:

$$R_{mxn} = \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,n} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ R_{m,1} & R_{m,2} & \cdots & R_{m,n} \end{bmatrix}$$

(2) u.item: movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |

The file is project information. Five fields: movie ID (IID), title of the movie, the movie released video time, release time, customers; 19 field behind the movie category list. The corresponding field for the 1 to show the film belongs to the category.

(3) u.genre: genre title | genre id

This paper mainly lists the project category name and the corresponding category code, namely, the 0-18 to represent the 19 categories, the order of the corresponding sequence of documents u.item.

(4) u*.base 和 u*.test

The dataset contains the u1.base~u5.base five training sets and the corresponding u1.test~u5.test five test sets. The training set and test set are randomly divided according to the proportion of 80% and 20%.

(5) u.user: user id| age | gender | occupation | zip code

This file lists the main information of the participating users, such as user UID, age, gender, occupation and zip code *etc*.

User score is 1,2, 5, 3, 4, 5 grades, the score is higher, show that the user is more like the movie, and vice versa, that is, 5 said the most like, 1 said the least favorite.

The experiment mainly uses the data to focus on the first four files, and then uses the data. The hardware configuration of the machine: Core Duo (TM) 2 CPU 2.00GHz Intel, RAM 2 GB, hard disk 500G. Running environment: operating system isWin7, development platform is MATLAB 2010.

3.2 Evaluation Criteria

The quality of recommendation is a decisive factor for the sustainable development of system: good recommendation quality can help attract new users in addition to keeping users' higher loyalty; on contrary, poor recommendation quality will lead to fewer users because of bad user experience.

Indicators for evaluating the quality of system recommendation are differing for different system targets. However, there're mainly two criteria: Statistical Accuracy Metrics and Decision Support Accuracy Metrics [14].

Mean absolute error, as a statistical precision method, is the most often used to measure the quality of recommendation. By calculating errors between system's predictive recommendation value and user actual evaluating value, it examines the accuracy of recommendation. Normally, people would get predictive recommendation value through training; then do testing with MAE; the smaller MAE value is, the higher the recommendation quality of system becomes.

Suppose in the testing dataset, the set of items rated by user U_i is $\{p_1, p_2, ..., p_N\}$. With the proposed recommendation algorithm, we can predict relative rating set is $\{r_1, r_2, ..., r_N\}$; and that MAE can be reached by the following equation:

$$MAE = \frac{\sum_{i=1}^{N} | p_i - r_i |}{N} \tag{5}$$

3.3 Experiment and Result Analysis

Experiment one: Observe the change tendency of MAE value when ItemNum and UserNum are given differently;

1. Experiment content

   (1)Fix ItemNum, observe MAE value change when UserNum increases from 10 to 50;
   (2)Fix UserNum, observe MAE value variation when ItemNum grows from 10 to 100.

2. The experimental results are shown in Figure 1.

3. Result analysis

When user's nearest neighbor UserNum is fixed, the system recommend accuracy raises along with the number of ItemNum; when ItemNum is fixed, the new algorithm's MAE value becomes smaller with increasing UserNum. When UserNum=50, ItemNum=100, the new algorithm gets the minimum MAE value, which is 0.7302.

4. Experimental conclusion

Item nearest neighbor determines the precision of predicting unrated items, which further affects the accuracy of acquiring the nearest neighbors. That is decisive to the accuracy of the final recommendation result. Hence, item nearest neighbor and user nearest neighbor both have big impacts on the quality of recommendation results.

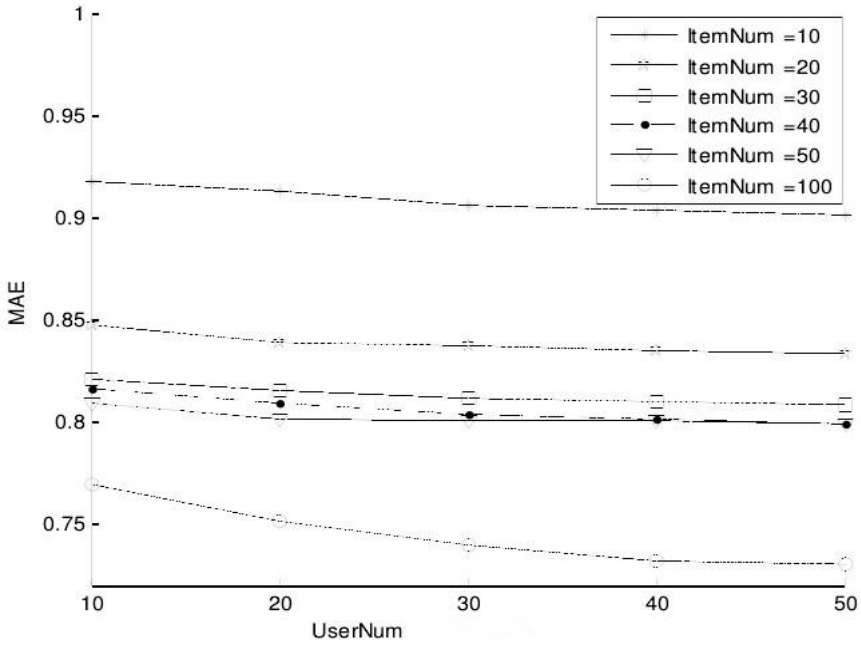

Figure 1. Comparison of Collaborative Filtering Recommendation Algorithms of Different Nearest Neighbors

Experiment two: comparison of the proposed algorithm and T-IC CF [15]algorithm

1 Experiment content

On the basis of identical dataset and training set and testing set at the same proportion, we compare MAE values with the method here and traditional collaborative filtering algorithm based on item category, by setting ItemNum as respectively 50 and 100. In Figure 1, C stands for the similarity calculation method based on cloud model, which is the proposed algorithm; T represents the conventional similar approach, *i.e.* T-ICCF algorithm.

2 The experimental results are shown in Figure 2.

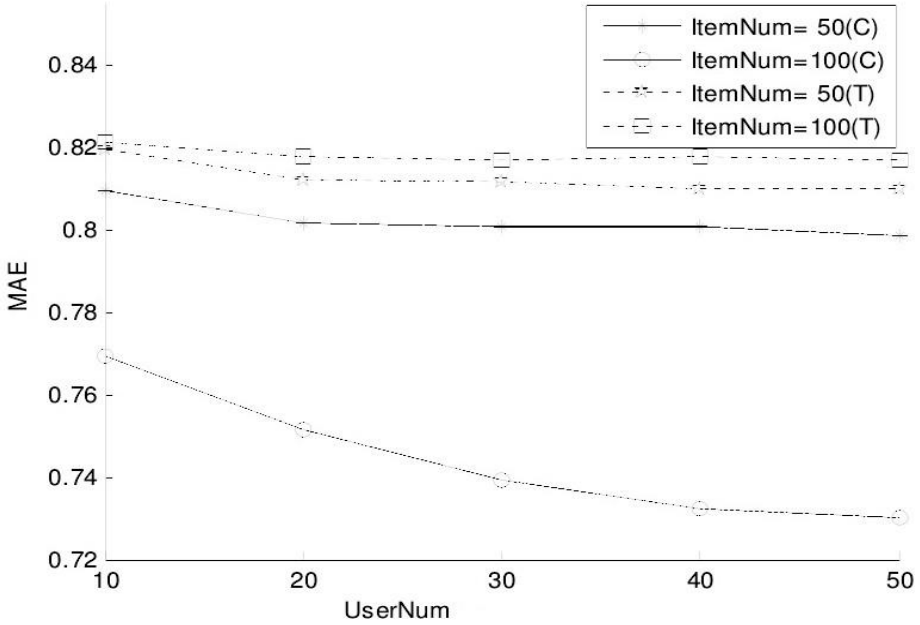

Figure 2. Compare with the T-IC CF Algorithm

3. Result analysis

(1) When ItemNum=50, T-IC CF algorithm got smaller MAE value than ItemNum=50; while the proposed method made smaller MAE value when ItemNum=100;
(2) No matter ItemNum=50 or 100, the method discussed here reached smaller MAE value than T-IC CF algorithm.

4. Experimental conclusion

When item classification is utilized, with the traditional similarity measuring method, when item's neighbor number is fewer, much better recommendation results are made due to strict matching object attribute; when that number is more, the similarity computing method based on cloud model can realize more accurate recommendation results. On the whole, the proposed solution reaches better recommendation effects than the collaborative filtering algorithm based on item classification.

Experiment three: observe the recommendation effect of the proposed method and peer one

1. Experiment content

Under the condition of identical dataset, training set and testing set at the same proportion, we compared the method in the paper with LICMCF algorithm and T-IC CF algorithm about the recommendation result.

2. The experimental results are shown in Figure 3.

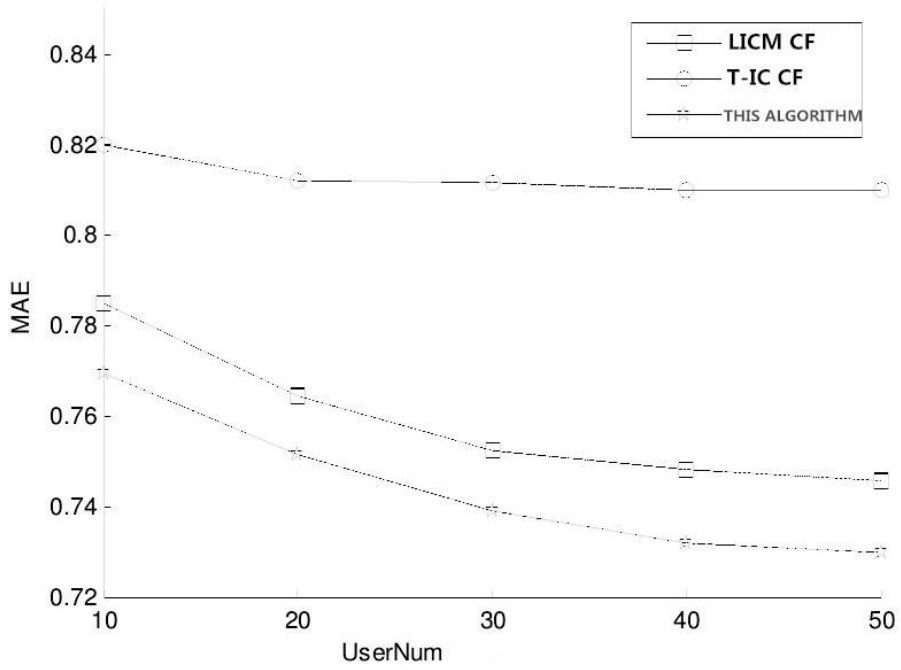

Figure 3. Comparison of MAE of Recommendation Algorithms

3. Result Analysis

When user neighbor is determined, MAE values of all three algorithms decrease along with the increasing item neighbors; moreover, MAE value of the discussed algorithm changed from 0.7695 to 0.7302, achieving much better recommendation results than the other two; when item neighbor is fixed, MAE values of the three method become smaller with more user neighbors. When there're over 50 user neighbors, their MAE values tend to keep stable.

4. Experimental conclusion

The collaborative filtering algorithm based on cloud model gets rid of the weakness of strict matching object attribute with the traditional approaches. Its effect is superior to the one based on item category. By integrating item classification and cloud model, the proposed algorithm improves the accuracy of searching nearest users, working better than the collaborative filtering algorithm based on the cloud model.

5. Conclusion

By introducing the idea of item classification, we can provide different interest recommendation sets to different users and thus the searched nearest interest neighbors are more accurate; by inputting the idea of cloud model, we can avoid the shortage of strict matching object attribute with the traditional similarity measurement methods, enhancing the recommendation quality of the system. Integrating item classification and the cloud model, the collaborative filtering algorithm possesses those merits. Besides it's found that the method calculates only the class to which user interest is added, instead of the entire data, which contributes to the system efficiency and scalability.

Acknowledgement

This work was supported by Mudanjiang Normal College science and technology research projects. No. QY2014002

References

[1] Sun Min. Collaborative filtering recommendation algorithm for improved user model. Chongqing University, 2012
[2] Lu Xindong. Research and application of collaborative filtering recommendation algorithm based on cloud model. North China Electric Power University, 2011
[3] Wang nianhong. Collaborative filtering recommendation algorithm based on cloud model. Computer system application, 2015,05:140-146.
[4] Hongying. The collaborative filtering recommendation algorithm based on user characteristics and cloud model, Jiangxi University of Science and Technology, 2014
[5] Xia Peiyong. Research on collaborative filtering algorithm in personalized recommendation technology, Ocean University of China, 2011
[6] Liu Qingwen. Research on the recommendation algorithm based on collaborative filtering. University of Science & Technology China, 2013
[7] Qi Lili. Research on Collaborative Filtering Recommendation Algorithm Based on MapReduce. Taiyuan University of Technology, 2014
[8] Huang Yang. Research on Collaborative Filtering Recommendation Algorithm Based on item clustering and preference categories. Zhejiang Sci-Tech University, 2014
[9] Wang Weijie. Research on the collaborative recommendation based on score prediction. East China Normal University, 2014
[10] Liu Fasheng, Hong Ying, collaborative filtering recommendation algorithm based on user characteristics and cloud model. Computer engineering and science, 2014, 06:1172-1176.
[11] Xiong Yu. Research on personalized service recommendation system for e-commerce in collaborative filtering. Electronic Science and Technology University, 2013
[12] Jiang Chong. Study on e-commerce recommendation system. Hunan: Central South University, 2009
[13] Wu Ting. Application and research of collaborative filtering technology in the electronic commerce recommendation system. Wuhan: Wuhan University of Technology, 2009
[14] Wen huiping. Based on item category similarity and the user interest in personalized recommendation algorithm research. Shanxi: Taiyuan University of technology, 2008
[15] Xiong Zhongyang, Liu Qin, Zhang Yufang *et al*. Collaborative filtering algorithm based on item classification. Computer application research, 2012,29 (2):493-496

Author

Dan Zhang. **S**he received her B.S degree from Harbin Normal University and received her M.S degree from University of Electronic Science and Technology of China. She is a lecturer at Institute of Engineering of Mudanjiang Normal University. Her research interests include software testing.

