



# Probability Matrix Factorization Algorithm for Course Recommendation System Fusing the Influence of Nearest Neighbor Users Based on Cloud Model

Jianguo Li, Chao Chang, Zuoxi Yang, Hailin Fu, and Yong Tang<sup>(✉)</sup>

South China Normal University, Guangzhou 510631, China  
ytang@m.scnu.edu.cn

**Abstract.** With the explosion of data on the online course website, getting the required course information quickly and accurately becomes more and more difficult. In this paper, probability matrix factorization algorithm for course recommendation system fusing the influence of nearest neighbor users based on cloud model is proposed. The proposed algorithm uses the cloud model to compute user similarity and integrates social information into the course recommendation. The experimental results show that the algorithm can improve the accuracy of course recommendation effectively.

**Keywords:** Course recommendation · Probability matrix factorization · Social network · Cloud-model · SCHOLAT

## 1 Introduction

With the rapid development of information technology, more and more online course platforms are emerging one after another and anyone can choose courses on the online course platform to start learning. Therefore, the learning resources show an explosive growth, which makes it impossible for users to obtain the required information quickly and accurately in the massive course information and resources [1]. As a web-based course website for scholars, SCHOLAT course platform also encounters such problems. Recommendation system can effectively alleviate the problem of information overload by studying user data, making personalized recommendation and recommending courses that meet the user's interest.

Collaborative filtering recommendation algorithm is one of the most widely used recommendation algorithms, mainly divided into memory-based collaborative filtering and model-based collaborative filtering [2]. David Goldberg et al. proposed the recommendation system Typestry based on collaborative filtering [3]. Paul Resnick et al. proposed GroupLens, an automated collaborative filtering recommendation system based on user ratings [4]. Therefore, collaborative filtering recommendation system is gradually applied to e-commerce and social networking sites. In real world scenarios, it is impossible for users to rate every item. The online course website also has the problem of data sparsity.

In the absence of sufficient statistics, it is difficult for the memory-based approach to provide accurate predictions. The model-based method can effectively alleviate the cold start problem by training the recommendation model with existing user information [5]. The matrix decomposition recommendation algorithm is widely used as a model-based method [6, 7, 8] because of its simplicity and high accuracy. Probability matrix decomposition [9] has achieved good results in various application areas since it is proposed.

The use of social information in social networking sites for recommendation, that is, social recommendation, has become a hot topic in the field of recommendation systems. Researcher Li [10] introduced co-author information as the user's associated information to improve the accuracy of recommendation. Researcher [11] use social tagging as a basis to solve the problem of recommendation. The course platform of SCHOLAT is different from the traditional course website. In addition to the elective function, it also has rich user social information and academic information. However, there is little research on using social information and academic information to recommend courses. In this paper, we will analyze the similarity of social information and the influence of users in social networks, and use the probability matrix decomposition method to recommend more accurate course for users.

## 2 Related Work

### 2.1 Similarity Computation of Nearest Neighbor Users

The methods of measuring user similarity mainly include cosine similarity, Pearson correlation coefficient and Euclidean distance. All three methods are based on the similarity computation of vectors, which is a strict match between object attributes.

- (1) Cosine similarity: User ratings are used as vectors in n-dimensional space, and user similarity is measured by the angle between the cosines of the two vectors. Given two vectors of attributes, the scores of user i and user j in n-dimensional space are vector  $\vec{i}$  and  $\vec{j}$  respectively, the cosine similarity between user i and user j is:

$$\cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \cdot \|\vec{j}\|} \quad (1)$$

- (2) Pearson correlation coefficient: Set  $X_i$  be the actual score of user X in item i,  $Y_i$  be the actual score of user Y in item i,  $\bar{X}$  be the average score of user X, and  $\bar{Y}$  be the average score of user Y, then user X and user Y's Pearson correlation coefficient is:

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (2)$$

- (3) Euclidean distance: Set two users  $X$  and  $Y$ ,  $X_i$  is the actual score of user  $X$  in item  $i$ , and  $Y_i$  is the actual score of user  $Y$  in item  $i$ , then the Euclidean distance of the two users is:

$$d(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (3)$$

## 2.2 Matrix Factorization

Because of its relatively low time and space complexity and high prediction accuracy, matrix factorization is widely used in recommender systems. The goal of matrix decomposition is to decompose the user-course elective matrix  $R$  into the user's hidden information feature matrix  $U \in R^{M \times K}$  and the hidden information and the hidden information feature matrix of the course  $V \in R^{N \times K}$  ( $K \ll \min(M, N)$ ),  $K$  is the dimension of the hidden information feature vectors.  $U$  and  $V$  are used to predict missing items in  $R$ , and the predicted values are compared for recommendation, that is:

$$R \approx UV^T = \hat{R} \quad (4)$$

In collaborative filtering recommendation algorithm, probabilistic matrix decomposition assumes the potential distribution of data, which makes up for the over-fitting problem of traditional matrix decomposition method and performs well in collaborative filtering algorithm.

## 3 Probability Matrix Factorization Algorithm for Course Recommendation System Fusing the Influence of Nearest Neighbor Users Based on Cloud Model

In order to reflect the influence of neighboring users on target users, this paper proposes a course recommendation system that integrates the influence of neighboring users and the decomposition of probability matrix.

### 3.1 Computation of User Similarity Based on Cloud Model

In this paper, a qualitative and quantitative transformation model named cloud model proposed by academician Deyi Li is used to compute the similarity between two users [12]. The cloud model uses  $E_x$  (expectation),  $E_n$  (entropy),  $He$  (superentropy) three parameters to characterize a fuzzy concept, so that the fuzziness of things and randomness organically combined. The model is defined as follows:

Definition: Set  $U$  be a fuzzy set and  $C$  is a qualitative concept on  $U$ . If the numerical  $a \in U$  is a random implementation on  $C$ , and  $A$  is a stable random with a tendency for the determinacy  $y: Y \rightarrow [0, 1] \forall a \in Y a \rightarrow y(x)$ . The distribution of all  $a$  in the fuzzy domain  $U$  can be called a cloud, and each numerical  $a$  in the cloud can be called a cloud droplet.

Where  $E_x$  denotes the expectation that a distributes on the fuzzy field U,  $E_n$  denotes the measurability of C, and He denotes the uncertainty of entropy, which is determined by the randomness and fuzziness of entropy.

If the cloud-model between users is similar, the three-dimensional vector formed by controlling the parameter expectation, entropy, and super-entropy generated by the cloud droplet should also be similar. In this paper, the above three parameters are obtained by the reverse cloud algorithm, and the user's three-dimensional vector is formed. The user's similarity is computed by the user's three-dimensional vector.

Let  $r_1, r_2, r_3, \dots, r_{n-2}, r_{n-1}, r_n$  be cloud droplet (rating). The computation steps of the three parameters of  $E_x, E_n$  and He for controlling cloud droplet generation are as follows:

Step1: Compute the sample average of the cloud droplet:  $\bar{r} = \frac{1}{n} \sum_{i=1}^n r_i$ ; Compute the first order absolute central moment of cloud droplet:  $\frac{1}{n} \sum_{i=1}^n |r_i - \bar{r}|$ ; Compute the variance of cloud droplet:  $S^2 = \frac{1}{n-1} \sum_{i=1}^n (r_i - \bar{r})^2$ .

Step2: Estimated value of  $E\hat{x} = \bar{r}$ ;

Step3: Estimated value of  $E\hat{n} = \sqrt{\frac{\pi}{2}} \times \frac{1}{n} \sum_{i=1}^n |r_i - E\hat{x}|$ ;

Step4: Estimated value of  $H\hat{e} = \sqrt{S^2 - \frac{1}{3}E\hat{n}}$ .

Using the reverse cloud algorithm, the user preference is represented by three parameters of the cloud computed from the user's original rating data. The cloud of three parameters of the user rating is called user characteristic vector and recorded as  $\vec{V} = (Ex, En, He)$ , where Ex reflects the user's comprehensive performance of each score. En reflects the user's concentration in the various ratings, that is, the measures of dispersion. And He be the stability of the entropy. User similarity based on cloud model is defined as follows:

Definition: Set the characteristic vector of two users be  $\vec{V}_i$  and  $\vec{V}_j$ ,  $\vec{V}_i = (Ex_i, En_i, He_i)$ ,  $\vec{V}_j = (Ex_j, En_j, He_j)$ . Their cosine angles are called the similarity of cloud i and cloud j.

$$\cos(\vec{V}_i, \vec{V}_j) = \frac{\vec{V}_i \cdot \vec{V}_j}{\|\vec{V}_i\| \cdot \|\vec{V}_j\|} \quad (5)$$

### 3.2 Recommendation System

PMF is a probabilistic linear model. The addition records of N course sets ( $I = \{I_j | j = 1, 2, 3, \dots, N\}$ ) by M user sets ( $H = \{H_i | i = 1, 2, 3, \dots, M\}$ ) constitute the observed scoring matrix R.  $R_{ij}$  denotes user i's selection of course j, the missing item or 0 item in R means that this course is not added, and 1 means that the course has been added. The PMF model We assume that the conditional distribution of obey the Gaussian distribution, and the gradient descent method is used to correct the variables. To prevent overfitting, the regularization term is added and iteratively continues until the algorithm finally converges (Fig. 1).

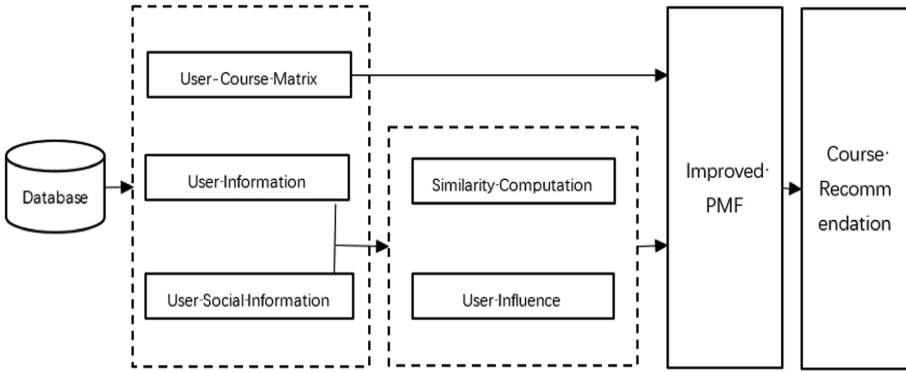


Fig. 1. Structure of proposed recommendation model

We suppose that the hidden information matrix of students-courses obeys Gauss prior distribution:

$$P(U|\delta_U^2) = \prod_{i=1}^M N(U_i|0, \delta_U^2 I) \tag{6}$$

$$P(V|\delta_V^2) = \prod_{j=1}^N N(U_j|0, \delta_V^2 I) \tag{7}$$

We suppose that the acquired student-elective scoring matrix of conditional probability also obeys the Gaussian priori distribution:

$$P(R|U, V, \delta_R^2) = \prod_{i=1}^M \prod_{j=1}^N [N(R_{ij}|g(U_i V_j^T), \delta_R^2)]^{I_{ij}^R} \tag{8}$$

Where  $I$  is the unit matrix,  $\delta_U^2, \delta_V^2$  are the variances of  $U$  and  $V$  distributions respectively.  $I_{ij}^R$  is the indicator function, and if the user  $U_i$  does not join the course  $V_j$ , then its value is 0, and if it is joined, its value is 1.  $g(x)$  maps the value of  $U_i V_j^T$  to the interval  $[0,1]$ . In this paper, we use  $g(x) = \frac{x-x_{min}}{x_{max}-x_{min}}$ , where  $x$  is the original data,  $x_{max}$  is the maximum of the original data set,  $x_{min}$  is the minimum of the original data set.

The posterior distribution function of the feature matrix can be found by Bayesian theorem:

$$P(U, V|R, \delta_R^2, \delta_U^2, \delta_V^2) \propto P(R|U, V, \delta_R^2) \times P(U|\delta_U^2) \times P(V|\delta_V^2) \tag{9}$$

Logarithmic processing of the above formula:

$$\begin{aligned} \ln p(U, V|R, \delta_R^2, \delta_U^2, \delta_V^2) = & -\frac{1}{2\delta_R^2} \sum_{i=1}^M \sum_{j=1}^N I_{ij}^R \left( R_{ij} - g(U_i V_j^T) \right)^2 \\ & -\frac{1}{2\delta_U^2} \sum_{i=1}^M U_i U_i^T - \frac{1}{2\delta_V^2} \sum_{j=1}^N V_j V_j^T \\ & -\frac{1}{2} \left( \left( \sum_{i=1}^M \sum_{j=1}^N I_{ij} \right) \ln \delta_R^2 + M K \ln \delta_U^2 + N K \ln \delta_V^2 \right) + C \end{aligned} \tag{10}$$

Maximizing the logarithm of a posteriori is equivalent to minimizing its following objective function:

$$f = \frac{1}{2\delta_R^2} \sum_{i=1}^M \sum_{j=1}^N I_{ij} \left( R_{ij} - g(U_i V_j^T) \right)^2 + \frac{\lambda_U}{2} \sum_{i=1}^M U_i U_i^T + \frac{\lambda_V}{2} \sum_{j=1}^N V_j V_j^T \quad (11)$$

In the above formula,  $\lambda_U = \frac{\delta_R^2}{\delta_U^2}$ ,  $\lambda_V = \frac{\delta_R^2}{\delta_V^2}$ .

The cloud model is used to compute the similarity between users, and a collection of neighboring users of each user is obtained. The influence of users is computed according to their academic information, academic degree, number of papers, number of projects, number of patents, number of works, etc., and the influence of nearby users is filled into the influence matrix  $D$  of nearby users.  $D_i \in [1, 2]$  be the rating of the user  $U_i$  in the influence matrix. If the user  $U_j$  is the user  $U_i$  neighbor user,  $D_{ij}$  equals the user's influence score, when the user  $U_j$  is not the user  $U_i$  neighbor user,  $D_{ij}$  equals 1. The rating contributions of different users were determined according to the influence of these users on target users. The more influential the users, the greater the proportion of the rating prediction of target users.

We use gradient descent method, using  $U_i$  and  $V_j$  as parameters. To reduce the computational complexity,  $\lambda_U = \lambda_V = \lambda$ .  $U_i$ ,  $V_j$  iterate over each iteration.  $U_i \leftarrow \left( U_i - \gamma \cdot \frac{\partial f}{\partial U_i} \right) D_i$ . Weighted control of each iteration by multiplying a similar user influence matrix. Where  $\gamma$  be the threshold. We get the hidden information characteristic matrix  $U$  and  $V$ , and then we can predict the reconstructed score matrix  $\hat{R}$  based on the eigenvector.

## 4 Experimental Setup and Analysis

### 4.1 Data Set

In this paper, a comprehensive teaching and scientific research collaboration platform developed by SCHOLAT research is used as an experiment. We use the data set provided by the SCHOLAT until March 16, 2018. A total of 43905 users and 1698 courses were recorded. User information included number of papers published, number of applications, landing score, dynamic score, academic score, academic degree, patents, number of teams joined, number of courses joined, number of friends joined, number of praise points, etc.

### 4.2 Evaluation Measures

In this paper, the root-mean-square error (RMSE) is used as the measure standard. The smaller the value of the RMSE, the higher the accuracy of the algorithm.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i \in m, j \in n} (r_{ij} - \hat{r}_{ij})^2} \quad (12)$$

Where  $r_{ij}$  is real relationship between the user and the course selection,  $\hat{r}_{ij}$  is the predictive relationship between the user and the course in the recommendation system.

### 4.3 Experimental Results and Analysis

We compare the proposed recommendation model with the traditional probability matrix decomposition method, the nearest neighbor user influence decomposition method using cosine similarity, the nearest neighbor user influence decomposition method using Pearson similarity, and the RMSE value of the nearest neighbor user influence decomposition method using Euclidean distance.

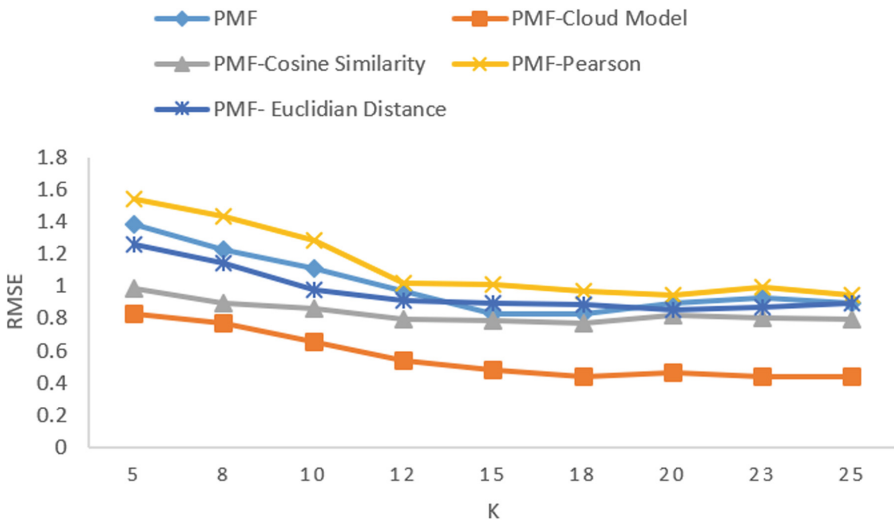
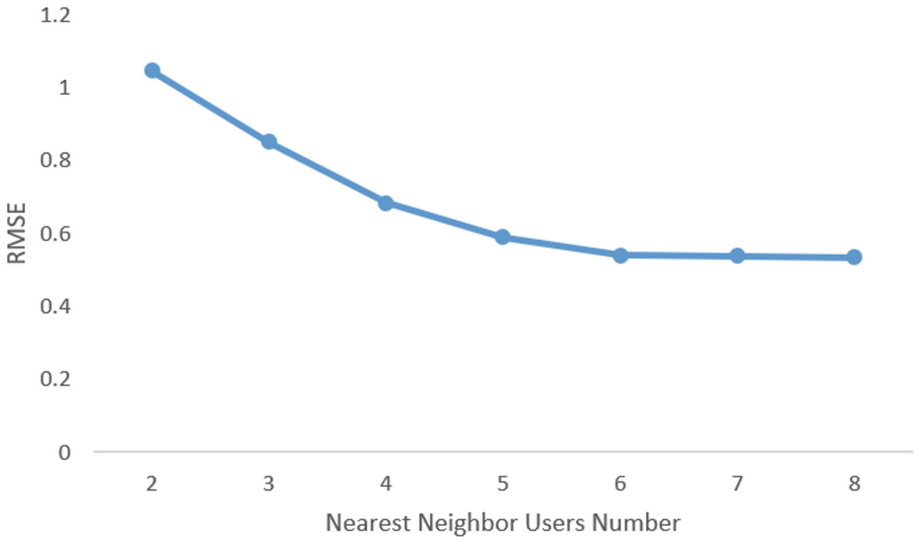


Fig. 2. The RMSE value polygon diagram of different algorithms when K takes different values

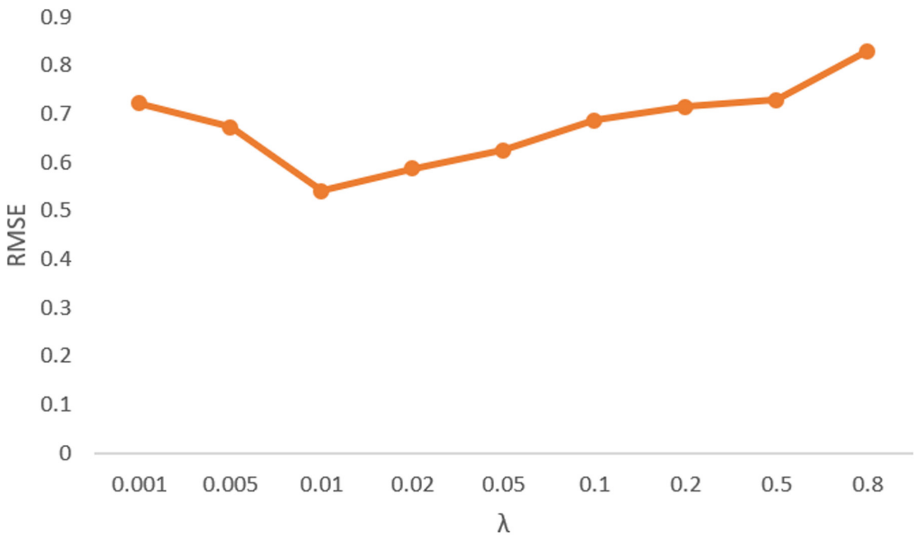
As can be seen from the above Fig. 2, the number of neighbors and  $\lambda$  of the parameters are both the best parameters. As the dimension K of hidden feature vector increases, the accuracy of all algorithm recommendations is improved. On the other hand, the problem of overfitting and increasing computational complexity may arise. In each dimension, the proposed algorithm is superior to the traditional PMF algorithm and other similar competition algorithms.

As the number of nearby users increases, the recommendation results gradually improve. RMSE values tend to stabilize after the number of nearby users exceeds 6. As the number of neighbor users increases, the algorithm will consume more time. Therefore, we choose neighbor users as 6 (Fig. 3).

From the above Fig. 4, we can see that the RMSE value is the smallest when  $\lambda = 0.01$ , and the precision of the algorithm begins to decrease as lambda continues to increase, so select  $\lambda = 0.01$  as the lambda parameter value of this recommendation system.



**Fig. 3.** The RMSE value of the algorithm in different neighborhood user numbers



**Fig. 4.** The RMSE value of the algorithm in different  $\lambda$

## 5 Conclusion

Aiming at the problem of information overload on the online course website, this paper proposed a course recommendation system that integrates the influence of neighboring users and the decomposition of probability matrix by studying the SCHOLAT course



data of the new course platform, and provides the course recommendation for the system users. Experiments show that this method can provide better and more accurate course recommendation for users than the traditional probability matrix decomposition method, effectively improve the efficiency of users' inquiry course, and thus provide better services for users.

Although this paper combines social information, it does not consider the time sequence of social information. In future research, the time stamp mechanism can be considered to fully analyze the influence time and scope of social information in different time sequence, so as to obtain the recommended results of time information.

**Acknowledgements.** This work is supported by the National Natural Science Foundation of China (No. 61272067), the Science and Technology Project of Guangdong Province (Nos. 2017 A040405057 and 2016A030303058), and the Science and Technology Program of Guangzhou, China (No. 201604046017).

## References

1. Xia, Z., Song, A., Fang, D., et al.: A collaborative filtering recommendation mechanism for cloud computing. *J. Comput. Res. Dev.* **51**(10), 2255–2269 (2014)
2. Chen, L., Chen, G., Wang, F.: Recommender systems based on user reviews: the state of the art. *User Mod. User-Adap. Inter.* **25**(2), 99–154 (2015)
3. Goldberg, D., Nichols, D., Oki, B.M., et al.: Using collaborative filtering to weave an information tapestry. *Commun. ACM* **35**(12), 61–70 (1992)
4. Resnick, P., Iacovou, N., Suchak, M., et al.: GroupLens: an open architecture for collaborative filtering of netnews. In: *ACM Conference on Computer Supported Cooperative Work 1994*, pp. 175–186. ACM, Chapel Hill (1994)
5. Pagare, R.A., Patil, S.: Study of collaborative filtering recommendation algorithm scalability issue. *Int. J. Comput. Appl.* **67**(25), 10–15 (2014)
6. Pham, T.A.N., Li, X., Cong, G., et al.: A general graph-based model for recommendation in event-based social networks. In: *International Conference on Data Engineering 2015*, pp. 567–578. IEEE (2015)
7. Bokde, D., Girase, S., Mukhopadhyay, D.: Matrix factorization model in collaborative filtering algorithms: a survey. *Procedia Comput. Sci.* **49**, 136–146 (2015). Icac
8. Song, Y., Zhuang, Z., Zhao, Q., et al.: Real-time automatic tag recommendation. In: *International ACM SIGIR Conference on Research and Development in Information Retrieval 2008*, vol. 6, pp. 515–522. ACM (2008)
9. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. *Adv. Neural. Inf. Process. Syst.* **20**(2), 1257–1264 (2007)
10. Li J, Xia F, Wang W, et al.: ACRec: a co-authorship based random walk model for academic collaboration recommendation. *International Conference on World Wide Web 2014*, pp. 1209–1214. ACM (2014)
11. Rendle, S., Schmidt-Thieme, L.: Pairwise interaction tensor factorization for personalized tag recommendation. In: *ACM International Conference on Web Search and Data Mining 2010*, pp. 81–90. ACM (2010)
12. Zhang, G.-W., Li, D.-Y., Li, P., et al.: A collaborative filtering recommendation algorithm based on cloud model. *J. Softw.* **18**(10), 2403–2411 (2007)