Cooperative Filtering Program Recommendation Algorithm Based on User Situations and Missing Values Estimation

Jian Dong[1], Ruichun Tang[2(✉)], and Geqiang Lian[1]

[1] College of Information Science and Engineering, Ocean University of China, Qingdao, China
[2] Ocean University of China, No.238, Songling Road, Qingdao 266100, Shandong Province, People's Republic of China
tangruichun@l26.com

Abstract. Aiming at the sparsity problem of cold start and user item matrix in TV and movie personalized recommendation, this paper presents an improved collaborative filtering recommendation algorithm based on user situations and missing values estimation (BUM) applied to smart TV service. First of all, the users are clustered according to the cold start conditions. Then the user similarity of the cold start and non cold start users is calculated, and the neighbor users are selected. For cold start users, we model user attributes by analyzing user scenarios, and select neighbor user by user similarity which defined by scenario dissimilarity. For non-cold start users, we insert the default value based on user preferences into supplement of user-item rating matrix to solve the sparsity, and then calculate the similarity to select neighbor users. Finally, the results are obtained by using the neighbor users through the CF scoring prediction algorithm to estimate the rating. The experimental results show that the proposed algorithm is effective.

Keywords: Context awareness · Imputation
Neighborhood-based collaborative filtering (CF)
Program recommender systems

1 Introduction

The development of entertainment industry, people are getting more and more entertainment, and the research of program recommendation system (PRS) has become a popular subject. Collaborative filtering (CF), one of the most successful recommendation technologies [1], has been applied to program recommendation. The problems of cold start and sparsity are always the key factor influencing the accuracy of CF recommendation system (RS). How to solve the two problems is the key to improve the recommendation effect.

The study of the cold start problem combines a user-based approach and an item-based approach. Li *et al*. proposed a hybrid CFUI method for user-item, which is item-based approach. Fill in the blank of the user-item rating data to supplement the required rating [2]. Tang et al. proposed a strategy of building a meta scene, which combines

Z. Shi et al. (Eds.): IIP 2018, IFIP AICT 538, pp. 247–258, 2018.
https://doi.org/10.1007/978-3-030-00828-4_25

different scenario strategies to form a presupposition scene, and then divides the "meta scene". A cold start recommendation algorithm based on multi attribute scoring matrix is proposed by Yin *et al*. It generates new neighborhood sets based on item or user properties, and uses singular vector decomposition (SVD) to provide convenience for cold start users [4]. Wang *et al*. proposed a prediction framework based on other users' ratings for the same items [5].

These studies by modifying the traditional CF framework, use the additional information other than the user evaluation data, data matrix reconstruction and prediction method based on machine learning, to solve the cold start problem. However, under cold start conditions, the best way to ensure the robustness of a collaborative filtering recommendation system is to improve the similarity measurement. In this paper, we model user attributes by analyzing user scenarios, and select neighbor users by user similarity which defined by scenario dissimilarity to predict the rating. This method effectively solves the problems in the above study.

Another problem of the CF system is that the actual user item matrix is sparse. In order to solve the problem of data sparsity in collaborative recommendation system, many scholars preprocess data by clustering method to improve the performance of CF algorithm. Chen *et al*. proposed a hybrid method combining both graph-summarization and content-based algorithms by a two-phase user clustering approach, which can recommend items according to user interests. With respect to other methods, the algorithm could generate better recommendation result in sparse datasets and cold-start scenarios [6]. Saveski *et al*. proposed LCE (Local Collective Embeddings) algorithm, which integrates item content information and user historical behavior information into a unified matrix decomposition, and combines matrix decomposition prediction accuracy and item content attribute information to overcome sparsity issues [7]; Bhasin *et al.* presented a novel component of a hybrid recommender system at LinkedIn, where item features are augmented by a virtual profile based on observed user-item interactions. It is a way to think about Collaborative Filtering with content features [8].

These studies alleviate the influence of data sparsity to the CF algorithm to a certain degree, and improve the accuracy of CF algorithm. However, some problems are exposed, such as the instability of the recommended quality, the algorithm is inefficient for the low sparsity user - item matrix. In this paper, the method of missing value interpolation is used to solve the sparsity. By interpolating the blank rating in the user-item matrix, we can reduce the data error in the stage of prediction calculation, ensure the accuracy of the selected neighbors with users similarity. The default value is determined by the analysis of user preferences, and the quality of the algorithm is guaranteed to be stable.

The rest of this paper is organized as follows: In Sect. 2, we introduces the collaborative filtering algorithm in the related work; in Sect. 3, we introduces the recommendation algorithm based on collaborative filtering recommendation with User Situations and Missing Values; in Sect. 4 the experimental results of the algorithm are presented and analyzed; Finally, we make a brief concluding remark and give the future work in Sect. 5.

2 Related Work

2.1 Cold Start and User Situations Analysis

Because of the inherent characteristics of cold start users, the rating matrix cannot be the main basis of similarity analysis for cold start users. Model analysis of user by user information is a good way.

Tang *et al*. presented a dynamic personalized recommendation algorithm which uses user profiles and item content to extend the co-rating relationships between ratings through each attribute. The ratings reflect similar user preferences and provide useful recommendations [9]. Alhamid *et al*. used context-aware advice to provide information and used social content and relevant tags and rating information to potentially consider contexts to personalize search content [10]. The model uses social tags to explore potential preferences, reflecting the collected contextual information. He also proposed a ranking algorithm for context-based items to bridge the gap between media resources, user personal and co-preference, and identified contextual information.

The above studies make full use of user information and establish a user model instead of the rating matrix to analyze the similarity, which effectively improves the accuracy of the recommendation system under the influence of scoring on the neighbor selection. However, the simple considering the user properties will cause the problem of low scalability of the algorithm. In this paper, we distinguish between user groups, scenario analysis is employed to analyse cold start users, ensure the scalability of the algorithm.

2.2 Data Sparse and Missing Values Estimation

Data sparsity is one of the most challenging issues in the recommendation technology. Because users tend to evaluate only a small portion of the item in the system, the user-item rating matrix is usually very sparse, the density of matrix is about 1%. In addition, this problem may cause the CF approach based on neighbor cannot find a neighbor, so it can't make a precise suggestion. In order to overcome this problem, many methods have been proposed in previous research. One of these methods is to fill in missing data through interpolation, such as default voting, smoothing method, and missing value data prediction. In this article, we use interpolation to solve the problem of data sparsity.

Default voting is a straightforward imputation-based method that assumes default values for those missing ratings, such as exploiting the average ratings by a small group of users as the default ratings for other items in order to increase the size of the co-rated item set [11]. Ma *et al*. proposed a method by using some machine learning algorithms to smooth all the missing data in the user-item rating matrix. Taking the confidence of interpolation into consideration, they only fill in the missing data when confidence exists. The result of this approach is better because it prevents poor imputation [12]. However, the EMDP algorithm they proposed treats all missing data equality, lead to less adaptability to other data sets. Zhu *et al.* proposed a non-parametric iterative interpolation method for mixed attribute datasets, which deduced the probability density of independent attributes by creating mixed kernels [13].

All this interpolation method can improve the accuracy of recommendation for sparse matrix recommendation system effectively, the algorithm complexity is insufficient, the performance of the algorithm is guaranteed, but for some cold start users, the preliminary forecast rationality value cannot be guaranteed, filling the sparse matrix with fixed value is seldom consider the difference of attributes between users or items, each user and item are different from the others. The method of equal treatment affects the accuracy of these users. In this paper, we consider the difference between cold start and sparsity, handle different users separately, and interpolate matrix with user preferences to improve the accuracy of algorithm.

3 Bum

In order to solve the influence of sparsity and cold start on recommender system, we redefine the weight matrix to define the framework based on Top-N recommendation, and we can also develop the recommendation based on user by learning user weight matrix. In this section, due to the success of social prediction recommendation, we will propose a sparse linear model based on user scenarios, the model is not only employ the user item rating to learn user weight matrix, also use the user's social information to improve the quality of Top-N recommendation.

Definition 1: Multidimensional Top-N recommendation: Given user item matrix R, which contains m users and n items, and social networks of users. The binary adjacency matrix is expressed as W, where $w_{ij} = 1$ if there is a social connection between u_i and u_j. The goal is to estimate recommendation ratings of all missing values in R for each user u and recommend N missing values with the highest ratings to users. The example of the Multidimensional Top-N recommendation is shown in Fig. 1.

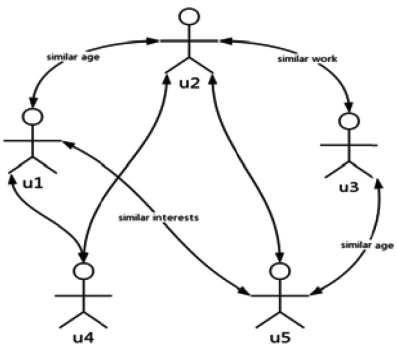


Fig. 1. User social relations: lines represents a similar attribute between users

As the Top-N based on item, first we propose a sparse linear model based on the user, It is assumed that the user u's recommendation rating r_{ui} for item i can be linearly represented by other users on the weight vector S_u, such as:

$$\widehat{r_{ui}} = r_i^T S_u \tag{1}$$

where r_i is the column vector of all user ratings of item i, and S_u is a sparse m dimension vector, which is composed of the weights between user u and all other users. Equation (1) can also be expressed in matrix form as follows:

$$\widehat{R}^T = R^T S \quad or \quad \widehat{R} = S^T R \tag{2}$$

where R represent a binary user-item matrix with m users and n items. Each row vector represents a recommendation rating for all items of user u, which is calculated as $\widehat{R}_u^T = S_u^T R$. S represents the weight matrix between users, and its *u-th* column corresponds to S_u Eq. (1). Obviously, it can be seen as a item based model, and for each user u, the weight column vector s_u in S can be learned as follows:

$$\begin{gathered} \min_{s_u} \frac{1}{2}||r_u - R^T s_u||_2^2 + \frac{\lambda_2}{2}||s_u||_2^2 + \lambda_1||s_u||_1 \\ \text{s.t } s_{uu} = 0 \end{gathered} \tag{3}$$

3.1 User Scenario Analysis

In the user data set, some users have very few scoring data relative to other users. The main reason for this problem is that such users are newly registered or rarely interact with other user group. Therefore, the calculation of the similarity of the group will be extremely inaccurate. The reference value of such users' ratings will become negligible. In order to find the nearest neighbor of such users, we should find other ways for user clustering to make the correct rating prediction.

Due to the characteristics of the cold-start users mentioned above, we are no longer using the user-item rating matrix, but analyzing the user scenarios to calculate the similarity of the nearest users of the active users.

The user scenario consists of basic user information, including age, gender and occupation. In this paper, we select the above three user information and use the triplet Cu to represent the concrete description, as in (4):

$$C_u = <Age, Gender, Occupation> \tag{4}$$

The value of the specific item in C_u, as in (5):

$$\begin{cases} Age \in (A = \{A_i | i = 1, 2, \ldots, 7\}) \\ Gender \in (G = \{0, 1\}) \\ Occupation \in (O = \{O_i | i = 1, 2, \ldots, 20\}) \end{cases} \tag{5}$$

where Age represents a collection of age attribute, which is composed of seven age groups, i.e., below the age of 18, 18–24, 25–34, 35–44, 45–49, 50–55 and 56 years of age or older; gender set is composed of two element 0 and 1, represents women and men respectively; occupation set contains 20 different types of occupations, such as teacher, doctor, engineer, student and so on.

After the abstract description of the user context information, the user is clustered to analyze. The cold start users are divided into several different user set, make the same set of users with similar scenario. We consider that the similarity is high, Users in different sets have different user scenarios, and we think they are not similar. Suppose the user set is $\mathrm{U} = \{u_i | \mathrm{i} = 1, 2, \ldots, \mathrm{m}\}$. It can be known from the user scenario definition that the gender of C_u in the triple is a binary variable, and the other attributes are nominal variables. Therefore, the traditional clustering algorithm can not calculate the similarity according to the user's scenario. In this case, the dissimilarity matrix may be used to describe the difference between user scenario data. The scenario dissimilarity $\mathrm{d}(C_i, C_j)$ may be calculated according to (6):

$$\mathrm{d}(C_i, C_j) = \frac{\sum_{v=1}^{l} m_{ij}^{v} \cdot n_{ij}^{v}}{\sum_{v=1}^{l} m_{ij}^{v}} \tag{6}$$

the user scenario is composed of l mixed variables, m_{ij}^{v} and n_{ij}^{v} are indicator functions. If the value of the v variable in C_i or C_j is missing, then $m_{ij}^{v} = 0$ otherwise $m_{ij}^{v} = 1$. When the values of the v variable in C_i and C_j are the same, then $n_{ij}^{v} = 1$ otherwise $n_{ij}^{v} = 1$.

The user scenario analysis in this article is aimed at all users, but the similarity analysis is only for the cold start user set. In order to select active users' K neighborhood users, referring to the Cosine similarity(COS), we used the scenario dissimilarity $\mathrm{d}(C_u, C_{u'})$ instead of user similarity to predict the rating as follows:

$$R^{*}(u, i) = \overline{R(u)} + \frac{\sum_{u' \in N(u)} \mathrm{d}(C_u, C_{u'}) \cdot \left(R(u', i) - \overline{R(u')}\right)}{\sum_{u' \in N(u)} \left|\mathrm{d}(C_u, C_{u'})\right|} \tag{7}$$

3.2 Default Value Supplementation Based on User Preferences

Compared with the user set that demands cold start process, for the set of users don't need to cold start, although these user set in terms of rating matrix is more rich than the cold start user set, but there are also a large number of blank data, which leads to the sparsity problem, so we also need to solve sparsity problem of these users. In this paper, we proposed a user preference based default value supplementation algorithm to solve the sparsity problem generated by such users.

We know the personal preferences of the user can be determined by the user rating of a class of things such as film. There are many classification of film: comedy, action,

love, the movies that be watched by the user through this classification method can be represented mathematically, as in (8):

$$X = \{x_1, x_2 x_3, \ldots\} \tag{8}$$

where X represents some kind of items, such as movie, music. It contains a specific classification of the X items.

We can define a user's specific preference for a class of items according to (9):

$$\bar{p} = \frac{\sum_{i \in X} R(u,i)}{|X|} \tag{9}$$

where |x| represents the size of the user rating data set for item X, and R (U, I) is a rating for a specific item, such as a rating for a movie. The value of P represents the average rating of user for a certain type of item.

We use the value of p to supplement the missing values in the user - item matrix. If the user does not rating a movie in a certain category, it shows that the user is not interested in such a movie, so the rating is still 0.

The rating matrix R of this kind of filling is changed to R′. The similarity calculation as in (10):

$$sim^*\left(u, u^{'}\right) = \frac{\sum_{i \in I\left(u,u^{'}\right)} R^{'}(u,i) \cdot R^{'}\left(u^{'}, i\right)}{\sqrt{\sum_{i \in I\left(u,u^{'}\right)} R^{'}(u,i)^2} \sqrt{\sum_{i \in I\left(u,u^{'}\right)} R^{'}\left(u^{'}, i\right)^2}} \tag{10}$$

Then according to the similarity, the K and the most similar users of the active users (that is, the neighbor users) are evaluated, as follows:

$$R^*(u,i) = \overline{R^{'}(u)} + \frac{\sum_{u^{'} \in N(u)} sim^*\left(u, u^{'}\right) \cdot \left(R^{'}\left(u^{'}, i\right) - \overline{R^{'}\left(u^{'}\right)}\right)}{\sum_{u^{'} \in N(u)} |sim^*(u, u^{'})|} \tag{11}$$

4 Experiments

4.1 Dataset

The dataset we experiment with is the popular benchmark dataset MovieLens, which include around 1 million ratings collected from 6040 users on 3900 movies. The ratings for the range of 1–5, user attributes include inherent user information such as age, gender and career. With these informations to build the user model for analysis. The incidental movie dataset contains information about the category of the movie. The sparsity of the rating matrix is 1–1000000/(6040 * 3592) = 0.9539.

4.2 Experimental Setup

4.2.1 Measurement

The general measurement metric of recommend system are as follows:

- Precision and Recall

The recall (Recall $= \frac{\sum_{u \in U} |R(u) \bigcup T(u)|}{\sum_{u \in U} |T(u)|}$) describes how many percentage of user-item rating records are included in the final recommendation list, and the precision rate (Precision $= \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}$) describes the proportion of the final recommendation list in the user-item rating record. These two indicators show the recommendation accuracy of a recommended algorithm. Therefore, it is important to select the appropriate K for the high accuracy of the recommended system. Of course, the accuracy of the recommended results is not particularly sensitive to K. As long as selected in a certain area, it can achieve good accuracy

- Coverage

Coverage (Coverage $= \frac{\left|\sum_{u \in U} R(u)\right|}{I}$) represents the coverage of a set of items recommended to the user by the recommendation system to the user's interest. The reduction of coverage because of the increase of popularity. With the increase of popularity, Recommendation system is more and more inclined to recommend popular items, so the recommendation for long tail items is less and less, lead to the decline of coverage.

4.2.2 User Clustering

The first stage of the proposed algorithm is to select the users who need cold start processing before the similarity operation is done to the users. By analyzing the number of user evaluations, we set a threshold θ. When the number of evaluated items $|R| < \theta$, they are classified as users who need cold start. The size of the threshold θ should be proportional to the expected value of most users, where θ is calculated according to (12):

$$\theta = \lambda \frac{\sum_{i=1}^{T} |R_i|}{T} \tag{12}$$

where T is the total number of ratings for active users u, and λ is the ratio coefficient, which is trained by specific data sets.

By classifying the user data sets above, the cold start processing users set U_{cs} for $|R| < \theta$ and other users set U_{ns} for $|R| > \theta$ can be obtained, as in (13):

$$u \in \begin{cases} U_{cs}, |R| < \theta \\ U_{ns}, |R| > \theta \end{cases} \qquad \mathrm{U} = U_{cs} \cup U_{ns} \tag{13}$$

4.2.3 Experimental Process

The main experimental process:

Input: user information set U, item information set I, rating data set R, threshold value theta, neighbor users number K.

Output: optimized the rating set R′, recommended list T

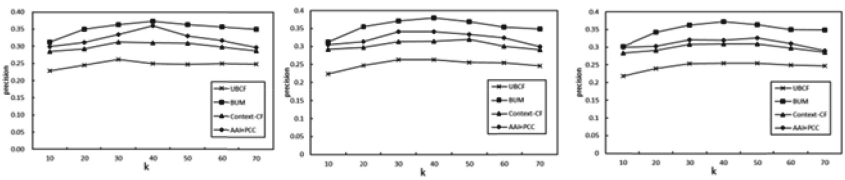
1. Divide the dataset into a training set and a test set.
2. The user set is divided into cold start users U_{cs} and other users U_{ns} by threshold value θ.
3. User scenario analysis for all user U, user set U_{cs} according to Eq. (6) is calculated by scenario dissimilarity.
4. For other users U_{ns}, the user preferences of active users are calculated by formula (9). Then the missing value in rating matrix of such users is supplemented, the supplement value is user preference, and the new scoring matrix R′ is obtained. Then the user similarity is calculated based on the rating matrix R′ and formula (10).
5. Recommended stage: Get K similar users U_{cs}' of the user set U_{cs} by using clustering algorithms, according to the scenario dissimilarity and rating record, get the recommended list T; for users U_{ns}, get K similar users of the user set U_{ns} by using rating matrix, according to Eq. (11) get the prediction rating of active user about the active item, and then get the recommended list of T.

4.3 Results and Analysis

Figures 2 and 3 show the accuracy and recall of four algorithms for the different values of the number of neighbor users K in the same experimental environment. We randomly selected three data sets of different sizes.

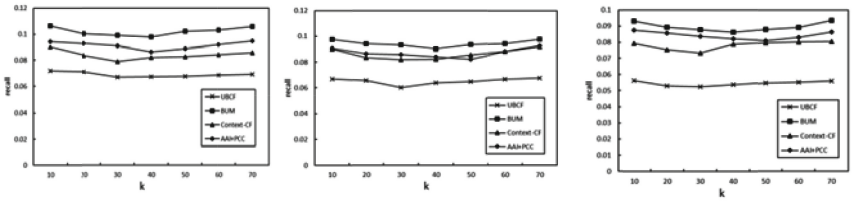
As shown in Figs. 2 and 3, we can see that:

(1) In determining the values of K, the accuracy and recall of BUM is always higher than the other three algorithms, of which BUM is the highest AAI + PCC followed by Context-CF and UBCF is the lowest. Analysis the reasons from the UBCF algorithm, as the traditional user based collaborative filtering algorithm, without considering the cold start and the sparsity problem emphatically, simple analysis of data from the user's point of view, similarity calculation based on the rating, and when the matrix is very sparse, its effect will be poor; For Context-CF, in order to solve the cold start and sparse problem, select similar users by using the user



(a) (b) (c)

Fig. 2. The relationship between the precision of different algorithms and the K value. (a) $|R| \approx 100000$ (b) $|R| \approx 500000$ (c) $|R| \approx 1000000$



(a) (b) (c)

Fig. 3. The relationship between the Recall of different algorithms and the K value (a) $|R| \approx 100000$ (b) $|R| \approx 500000$ (c) $|R| \approx 1000000$

attribute information to calculate similarity, finally predict rating for results, although UBCF partly solves the cold start and sparse problem, but too much use of user attributes, makes the user's rating data and film itself contains information not functioning properly, the recommendation results is lower than BUM and AAI + PCC; AAI + PCC is used for data interpolation to solve the problem of sparse matrix, considering users interested in the item information for dynamic interpolation, this algorithm has achieved good results, but for cold start users, because of the sparsity of users interest, for this kind of user recommendation effect is not ideal, and affect the overall prediction accuracy of the recommendation.

(2) For the selection of K, we can see from the accuracy of BUM algorithm, the higher the K value is between 0–40, the higher the accuracy are, and the maximum is achieved at K = 40, which indicates that K = 40 can get the best recommendation effect under the current experimental environment.

(3) Under different size and sparsity of data sets, the algorithm can still maintain a relatively stable recommendation precision. We can see that in different sizes of data sets, the precision and recall of the other three algorithms have been significantly changed. For example, in $|R| \approx 100000$, the precision and recall of AAI + PCC are higher than that of Context-CF, but the recall of AAI + PCC and Context-CF is similar in the case of $|R| \approx 500000$. For BUM algorithm, it can maintain the highest precision and recall in three cases. This shows that the BUM algorithm has excellent performance for different size datasets, and proves the stability of the algorithm.

Since the θ value of the cold start user is distinguished by the value of λ, we have studied the effect of the value of λ on the recommended performance. Figure 4 shows the impact of λ when K = 40, we added a new indicator named coverage, it is a widely used measurement metrics for recommend system evaluation. As shown in Fig. 5: The best results can be achieved when $\lambda = 0.3$, the accuracy and recall is the highest, and for coverage with theta increase its coverage is gradually reduced, this is because the BUM algorithm solving sparse problem for non cold start users through interpolation of user item matrix, The interpolation value is on behalf of the user's interest in certain items, So the higher the lambda value, the less users of cold start process, represents the user interest in the smaller range.

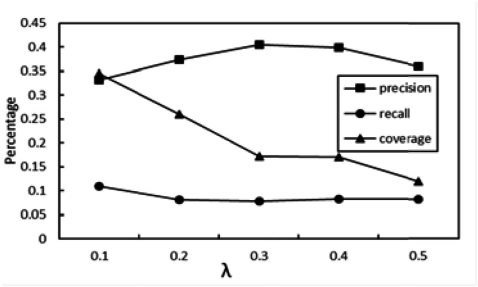


Fig. 4. Impact of λ

5 Conclusion

This paper presents a collaborative filtering recommendation algorithm based on user scenario and interpolation of intelligent program, the main work is as follows: At the stage of similarity computation, for cold start users (here refers to the user rating is very scarce or even no user), by analyzing the attributes of the user itself, such as age, gender, occupation, abandon the ratings of user attribute analysis, select the neighbor users by using scenario dissimilarity, and then recommend the prediction; for non cold start users, by using missing data interpolation to solve the sparse problem in user-item matrix, the interpolation value is calculated by the interest of users for item categories which rating value is missing, after the matrix interpolation, select neighbor users by calculating the similarity. and finally a recommendation is made by rating prediction. Compared with other algorithms, the algorithm proposed in this paper combines the cold start and sparsity solution, competition data indicate that the proposed algorithm is effective.

Acknowledgment. This work is supported by national key research and development plan under Grant no. 2017YFC0806205, CERNET Innovation Project under Grant no. NGII20160116.

References

1. Kim, E., Pyo, S., Park, E., et al.: An automatic recommendation scheme of TV program contents for (IP)TV personalization. IEEE Trans. Broadcast. **57**(3), 674–684 (2011)
2. Li, Y., Lu, L., Li, X.: A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce. Expert Syst. Appl. **28**(1), 67–77 (2005)
3. Tang, L., Jiang, Y., Li, L., et al.: Ensemble contextual bandits for personalized recommendation. In: ACM Conference on Recommender Systems, pp. 73–80. ACM (2014)
4. Yin, H., Chang, G., Wang, X.: A Cold-start recommendation algorithm based on new user's implicit information and multi-attribute rating matrix. In: International Conference on Hybrid Intelligent Systems, pp. 353–358. IEEE (2009)
5. Wang, J., Vries, A.P.D., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion, pp. 501–508 (2006)

6. Chen, K.H., Han, P.P., Wu, J.: User clustering based social network recommendation. Chin. J. Comput. **36**(2), 349–359 (2013)
7. Saveski, M., Mantrach, A.: Item cold-start recommendations: learning local collective embeddings. In: ACM Conference on Recommender Systems, pp. 89–96. ACM (2014)
8. Liu, H., Goyal, A., Walker, T., Bhasin, A.: Improving the discriminative power of inferred content information using segmented virtual profile. In: ACM Conference on Recommender Systems, pp. 97–104. ACM (2014)
9. Tang, X., Zhou, J.: Dynamic personalized recommendation on sparse data. IEEE Trans. Knowl. Data Eng. **25**(12), 2895–2899 (2013)
10. Alhamid, M.F., Rawashdeh, M., Dong, H., et al.: Exploring latent preferences for context-aware personalized recommendation systems. IEEE Trans. Hum. Mach. Syst. **46**(4), 615–623 (2017)
11. Johnson, R.A., Bhattacharyya, G.: Statistics: principles and methods. J. R. Stat. Soc. **43**(1), 922–925 (2006)
12. Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–46. ACM (2007)
13. Zhu, X., Zhang, S., Jin, Z., et al.: Missing value estimation for mixed-attribute data sets. IEEE Trans. Knowl. Data Eng. **23**(1), 110–121 (2010)