A Collaborative Filtering Algorithm Based on SVD and Trust Factor

Jianfang Wang, Pengfei Han, Yanling Miao and Fengming Zhang
College of Computer Science and Technology, Henan Polytechnic University Jiaozuo, China
wangjianfang@hpu.edu.cn,1060003755 @qq.com, 1337313091@qq.com, 674033623@qq.com

Abstract. At present, most collaborative filtering algorithms use similarity as a criterion. In order to alleviate problems of cold start and sparsity in recommender system, a Collaborative Filtering Algorithm Combined with the Singular Value Decomposition (SVD) and Trust Factors (CFSVD-TF) is presented. Further mining data features, we use the SVD to mining data features to gain the implicit Items feature space, then the items-based similarity are computed by using the improved cosine similarity. The trust factor is integrated into the similarity space to generate the computable trust model. Finally, to evaluate the proposed CFSVD-TF approach, the accuracy of the CFSVD-TF algorithm has significantly improved than the traditional CF algorithm in MovieLens datasets.

Keywords: Collaborative filtering algorithm, singular value decomposition, trust factor, nearest neighbor.

1. Introduction

The recommender system is an effective technique to solve the information overload problem [1]. Collaborative filtering in recommender system is one of the most successful techniques and based on users' predilection for items behavior. Find the similarity among users and items to recommend the item to target user [2]. But there are often many new users and many users do not have records ever. So the problems of sparsity and cold start have become a big challenge in recommender system. Therefore, the single calculate users' similarity or items' similarity always not accurate to get recommender results.

To solve data sparsity, usually using the following methods:

(1) Data smoothing technology, which fills in the default values for items that not yet be evaluated by users. For example, use other people who have rated the item, fill in the items with average ratings [3]. The other way, clustering user information and then use the same kind users' average ratings to fill in the items [4].

(2) Data dimensionality reduction, which reduces data from high dimension to low dimensional space. It mainly concludes PCA (Principal Component Analysis) and SVD (Singular Value Decomposition) [6]. SVD is an effective technology to reduce the data dimension and has the ability to automatically extract important features of matrices. Combining SVD and Collaborative filtering be used in Netflix Prize with rapid promotion [6]. The algorithm of NSVD and NSVD2 be proposed based on [6] and added users' and items' bias information [7].

The traditional collaborative filtering algorithm only takes the similarity between users or items into account, but the trust between users or items also is an important factor in the collaborative filtering algorithm which based on trust. It effectively improves the recommendation accuracy. A new algorithm is proposed, which utilize data mining algorithm to fill the sparse rating matrix, use filled matrix to compute the similarity between users and take the trust into account [8]. The trust between users also is proposed in [9].

In view of the above situation, we present a new recommendation method which combines the Trust Factors and SVD (Singular Value Decomposition), named CFSVD-TF (a Collaborative Filtering Algorithm Combined with the Singular Value Decomposition (SVD) and Trust Factors). Firstly, we use SVD to get the implicit feature space of the project, use the adjusted cosine similarity to compute the similarity between items and generate the temporary neighbor set. Then, use the trust factor to build a trust model and take it into the similarity space to provide better recommendation

results. At last this new method makes the recommendation on MovieLens. The experimental results show that this method not only effective in dimension reduction, but also change the style that prediction results only depend on the similarity between items, and provides the trust between items also is an import factor for prediction.

The remainder of the paper is organized as follows. Section 2 contains some preliminary and formally defines the problem solved. In Section 3, we present our CFSVD-TF method in four parts: feature space, two stages of kNN selection, trust factor and rating prediction. In Section 4 we discuss our experimental settings and analyze the results on real world Movie-Lens Dataset. Finally, Section 5 concludes the paper.

2. Preliminary and Related Work

Assume $\boldsymbol{U} = \{u_p | p \in \{1,\ldots,\mathrm{m}\}\}$ is the set of users, $\boldsymbol{S} = \{s_i | i \in \{1,\ldots,\mathrm{n}\}\}$ is the set of items, $\boldsymbol{R} = \{r_{p,i} | i \in \{1,\ldots,\mathrm{n}\}, p \in \{1,\ldots,\mathrm{m}\}\}$ is the rating matrix where $r_{p,i}$ represents the rating of s_i given by up, the range of $r_{p,i}$ is [1,5].

Table 1 show the rating matrix.

Table 1 User-Item rating matrix

	s_1	s_2	...	$s_{\mathrm{n-1}}$	s_{n}
u_1	5	0	...	5	0
u_2	0	0	...	0	4
u_3	2	0	...	0	0
...	0	0	...	0	0
$u_{\mathrm{m-1}}$	2	0	...	0	0
u_{m}	0	4	...	0	1

If there are not have any user ever rating for the item, we set it to be zero, like formula (1).

$$r_{i,p} = \begin{cases} r_{i,p}, & rating \\ 0, & not-rating \end{cases} \tag{1}$$

2.1 Singular Value Decomposition

In data mining, SVD as a matrix decomposition algorithm, it generates a low-rank matrix to approach the original matrix [10]. The SVD of matrix A can be defined as formula (2).

$$\boldsymbol{A} = \boldsymbol{U} * \boldsymbol{S} * \boldsymbol{V}^{\boldsymbol{T}} \tag{2}$$

Where $\boldsymbol{U} \in \mathrm{R}_{\mathrm{m*m}}$, $\boldsymbol{V} \in \mathrm{R}_{\mathrm{n*n}}$, $\boldsymbol{S} \in \mathrm{R}_{\mathrm{m*n}}$. Matrix $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrix, and its' column vector of matrixes $\boldsymbol{AA}^{\boldsymbol{T}}$ and $\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}$ feature vector respectively. Matrix $\boldsymbol{S}$ = diag, r is the rank of matrix $\boldsymbol{R}$. σ_r is the singular value of matrix $\boldsymbol{R}$, its' value is the average of eigenvalue by $\boldsymbol{AA}^{\boldsymbol{T}}$ or $\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}$. Therefore the effective dimensions of these three matrices are $m*r$, $r*r$ and $n*r$ respectively.

The SVD can provide the best approximation for the original matrix $\boldsymbol{A}$ by multiplying three matrices [11]. Firstly, get a new diagonal matrix by simplifying the matrix $\boldsymbol{S}$ with the largest singular value of k, where $k < r$. Then we get the simplified matrix $\boldsymbol{U_k}$ and $\boldsymbol{V_k}$ by deleting columns of matrix U and V. The simplified formula is shown in (3).

$$\boldsymbol{A_{red}} = \boldsymbol{U_k} * \boldsymbol{S_k} * \boldsymbol{V_k^T} \tag{3}$$

This is the closest unitarily invariant norm with the k bit approximation to the original matrix.

2.2 Similarity Calculation

Cosine similarity and adjusted cosine similarity are usually be used in collaborative filtering recommendation algorithm to estimate similarity [12]. The formula of cosine similarity as shown in formula (4).

$$sim(p,q) = cos\left(\vec{\boldsymbol{P}}, \vec{\boldsymbol{Q}}\right) = \frac{\vec{\boldsymbol{P}} * \vec{\boldsymbol{Q}}}{\vec{\boldsymbol{P}} \times \vec{\boldsymbol{Q}}} = \frac{\sum_{s \in S_{pq}} r_{p,s} \times r_{q,s}}{\sqrt{\sum_{s \in S_p} r_{p,s}^2 \sum_{s \in S_q} r_{q,s}^2}} \tag{4}$$

Where *sim* (*p,q*) is the similarity of u_p and u_q, which ranges from−1 to 1. $\vec{P}$ and $\vec{Q}$ are ratings vector of items by u_p and u_q, $\boldsymbol{S_p}$ and $\boldsymbol{S_q}$ are the ratings set by u_p and u_q, $\boldsymbol{S_{p,q}}$ is the common ratings set by u_p and u_q, $r_{p,s}$ and $r_{q,s}$ are the ratings on item *s* by u_p and u_q respectively.

Different users have different scoring rules, some users tend to high rating, but others tend to low rating. Cosine similarity did not take this into account, so we use adjusted cosine similarity to compute the similarity of users in recommender systems. It eliminates the influence of different users' scoring habits by minus the average of user ratings. The formula of adjusted cosine similarity as shown in formula (5).

$$sim(p,q)=\frac{\sum_{s\in S_{p,q}}\left(r_{p,s}-\overline{R_P}\right)\left(r_{q,s}-\overline{R_Q}\right)}{\sqrt{\sum_{s\in S_p}(r_{p,s}-\overline{R_P})^2\sum_{s\in S_q}(r_{q,s}-\overline{R_Q})^2}} \tag{5}$$

Where $\overline{R_P}$ and $\overline{R_Q}$ are the average ratings by up and u_q respectively.

3. Proposed CFVSD-TF Method

Traditional collaborative filtering only takes the similarity of users or items as the influencing factor to generate the neighbor set and recommend items to target user. In our daily life, we not only regard similar users as neighbors set or regard similar items as be recommended items set, but the item in the whole items set whether has a great word-of-mouth also affect users' decision-making, that is also to say the similarity between items and the degree of trust are important factors to affect the prediction accuracy.

3.1 Feature Space

Firstly, the zero rating items in the original score matrix $\boldsymbol{R}$ is replaced by the mean value of the correlation column. Then normalization of every row of the matrix to the same length to replace $\boldsymbol{R_{u,i}}$, $\boldsymbol{R_u}$ is the average score of the related columns. Finally we use SVD to get implicit feature space of items.

3.2 Two Stages of KNN Selection

Categorization algorithm of kNN (K-Nearest Neighbor) is one of data mining algorithms. Given a training dataset, for a new input instance, in the training dataset to find the *k* nearest neighbors to the instance, if most of these *k* neighbors belong to a certain class, then the input instance belong to this class. The algorithm of kNN is used twice in the CFSVD-TF to find the nearest neighbors. The first time, we use kNN to find the k (k=10, 20, 30) nearest neighbors in the items similarity matrix which generates by adjusted cosine similarity in items feature space, so we get the items neighbors set $\boldsymbol{Q_i}$. The second time is to integrate the items trust factor into the similarity space and search for the nearest neighbor again.

3.3 Trust Factor

In collaborative filtering, the items' degree is trusted in the user-item rating matrix, be named as the trust factor. It includes two parts: global trust and local trust.

Global trust, the credibility of a single item in all items.

The global trust of project i is represented by $\boldsymbol{T_i}$ ($\boldsymbol{0\le T_i\le 1}$). $\boldsymbol{T_i}$ can be computed as formula (6).

$$\boldsymbol{T_i}=\frac{2\left(1-1/\ln\left(f_i+2\right)\right)*\left(1-1/\ln\left(q_i+3\right)\right)}{2-1/\ln\left(f_i+2\right)-1/\ln\left(q_i+3\right)} \tag{6}$$

Where f_i is the number of users evaluated by item *i*. q_i is the number that item i be as the neighbor for other items. The number of q_i can get in section 3.2.

Local trust, the trust value between two items, if the similarity between the two items is higher, the trust value of the two items will be higher. So the local trust relies on the global trust and similarity. Combining the formula (5) and (6), the local trust can be computed as (7).

$$\boldsymbol{T_b}\left(\boldsymbol{P_a}\right)=\frac{2*sim(a,b)*\boldsymbol{T_a}}{sim(a,b)+\boldsymbol{T_a}} \tag{7}$$

Where $\boldsymbol{T}_b(\boldsymbol{P}a)$ is the local trust value from item *b* to =. *sim*(*a,b*) is the similarity between item *a* and item *b*. Ta is the global trust value of item *a*. Attention, the similarity between item *a* and item *b* is the

same as between item *b* and item *a*, that is, $sim(a,b)=sim(b,a)$. But the local trust between item *a* and item *b* is different with between item *b* and item *a*, that is, $T_a \neq T_b(P_a)$.

3.4 Rating Prediction

We can get the prediction score matrix $\boldsymbol{R_{pred}}$, through predict user *u*'s rating on item i. the predict value can be computed as the formula (8).

$$pr_{u,i} = \frac{\sum_{k=1}^{l} T_i(P_k) * \left(rr_{ui} + \overline{r_u}\right)}{\sum_{k=1}^{l} \left|T_i(P_k)\right|} \tag{8}$$

Where *l* is the number of item *i*'s neighbors which get with kNN. rr_{ui} is the user *u*'s rating on the item i in the matrix A_{red} which by dimension reduction through SVD. $\overline{r_u}$ is the average rating of user *u*.

3.5 Implementation of CFSVD-TF

The arithmetic idea of CFSVD-TF is: firstly use SVD to get the items' feature space, and use adjusted cosine similarity to compute the similarity between items, then get the temporary neighbors set by kNN, at last the trust factor is added.

Input: the origin rating matrix $\boldsymbol{R}$.

Output: the prediction rating matrix $\boldsymbol{R_{pred}}$.

S1: we can get the normalized matrix $\boldsymbol{R_{norm}}$ by filling the original rating matrix $\boldsymbol{R}$ with the items' average rating if the rating is zero, and get the matrix $\boldsymbol{U}$, $\boldsymbol{S}$ and $\boldsymbol{V}$ by using SVD on the matrix $\boldsymbol{R_{norm}}$.

S2: simplify the matrix $\boldsymbol{S}$ to k dimension to get the matrix $\boldsymbol{S_k}$ (k <r, rank ($\boldsymbol{R_{norm}}$) = r). Similarly, we can get the matrix $\boldsymbol{U_k}$ and $\boldsymbol{V_k}$ by simplify the matrix $\boldsymbol{U}$ and V respectively. $\boldsymbol{R_{red}}=\boldsymbol{u_k}*\boldsymbol{sk}*\boldsymbol{v^T_k}$, square roots of $\boldsymbol{S_k}$ to get $\sqrt{\boldsymbol{S_k}}$ and then get the items' implicit feature space $\sqrt{\boldsymbol{S_k}} * \boldsymbol{V^T_k}$.

S3: in feature space use formula (3) to get the similarity between item i and *j*.

S4: get items neighbors set $\boldsymbol{Q_i}$ by kNN and set $k = 20$, then get the number of q_i according $\boldsymbol{Q_i}$, get f_i by traverse through the original matrix $\boldsymbol{R}$.

S5: get $\boldsymbol{T_i}$, the global trust value of item *i*, by formula (6). Then use the value to fill matrix, use formula (7) to compute the local trust value between items.

S6: get the nearest neighbor set by kNN.

S7: predicting the item rating by formula (8).

S8: use algorithm of Top-N, recommending to the target user with N items which have high prediction rating.

4. Experiments

4.1 Dataset and Environment

In order to evaluate the performance of CFSVD-TF recommendation approach, we utilize the MovieLens 100k dataset from Minnesota University of America. Use formula (9) to evaluate the sparsity of matrix.

$$S_{lv} = 1 - \frac{N_{ui}}{m * n} \tag{9}$$

Where S_{iv} is the level of sparsity, $\boldsymbol{N_{ui}}$ is the number of items which be rated by user, *m* is the numb er of users, and *n* is the number of items.

The dataset consists of 100,000 ratings (1-5) from 943 users on 1682 movies and each user has rated at least 20 movies. If a movie is rated as one star, it means the user doesn't have a preference for the movie. If a movie is rated as five stars, it means the user like it. The more stars item has the user more preference the item. According to formula (9) we can get the data sparsity value is 1-100000 / (943*1682) = 0.9370.

Dataset be divided into five datasets (u1~u5), in these experiments, we randomly abstract 80% of ratings in the matrix as training sets, and use the 20% remained for testing, utilize 5-fold cross-valida tion to experiments.

4.2 Evaluation Metrics

As used in the most recent research papers, we use the RMSE (Root Mean Squared Error) to measure the error in a recommendation, The smaller the value of RMSE, the more precise a recommendation. The metric RMSE is defined as formula (10).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(P_i - r_i)^2}{N}} \tag{10}$$

Where $\{P_1,P_2,...,P_N\}$ means the predicting ratings of users on items, $\{r_1,r_2...,r_N,\}$ means the real ratings of users on items.

4.3 Results and Analysis

1) Distribution and selection of similarity: To get a better result, we compute the similarity between items by cosine similarity and adjusted cosine similarity respectively. The value distribution as Fig.1, as shown in this figure, the value distribution of cosine similarity is uniform, and adjusted cosine similarity has better performance on personality. The value of cosine similarity about 89.10% in range [0.0, 0.6], it's too scattered. But the value of adjusted cosine similarity about 80.15% in range [0.0, 0.4]. Because the users' rating on the item subtracts the users' average rating on items, can balance the problem of different user follow different rating scales. That is, users' personalized choice. Therefore we utilize adjusted cosine similarity to ensure get the high-quality prediction.

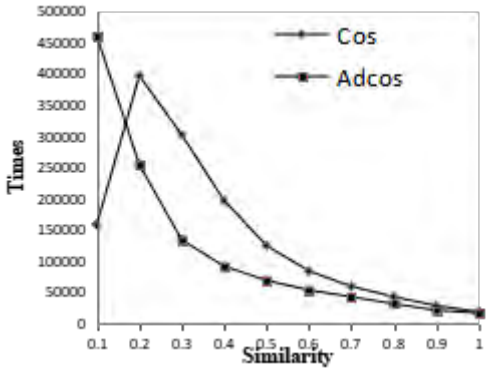

Fig. 1 Distribution of two kinds of similarity

2) Distribution and analysis of global trust: The distribution of trust factor f_i as shown in Fig. 2. In MovieLens 100k, the number of item be rated by users in [0, 500], about 67.8% in [0, 50], the number of remained items be valued are distributed in other intervals. That is, the trust factor f_i can represent the individual characteristics of a single item in whole items.

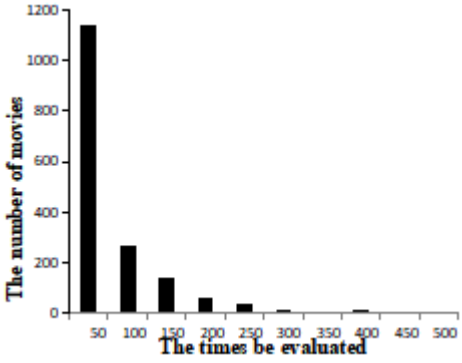

Fig. 2 Distribution of f_i

The Fig. 3 shows the result of the first use kNN, the number of the active items as others' neighbors distributed on [0, 800]. On [0, 50], when k=10, 20 and 30, the number of neighbors is 1603, 1522 and 1469 respectively. With the increase of k, the distribution of the trust factor q_i becomes uniform. According to the final result of the experiment, we get the best RMSE at k=20. When k=20, the trust factor of q_i are distributed on [0, 750], almost 10% of items are distributed on [50, 750]. That is, the trust factor of q_i is an important factor for global trust.

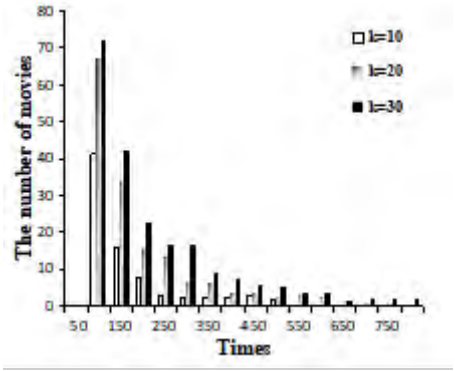

Fig. 3 Distribution of q_i

3) Distribution and analysis of trust: The Fig.4 shows the distribution of trust under different number of neighbors. Compare with the distribution of similarity, the distribution of trust values is more uniform, all intervals are distributed except [0, 0.1] and [0.2, 0.3]. About 88.88% of them are distributed on [0.4, 0.8]. In recommender system, each item gets a trust value by utilized trust algorithm, 90% of trust value between items distribution on [0.4, 0.8], but only 13.84% of similarity between items distribution on [0.4, 0.8].

Therefore trust factor and similarity are two completely different factors, and it is feasible to utilize the trust factor into collaborative filtering algorithm.

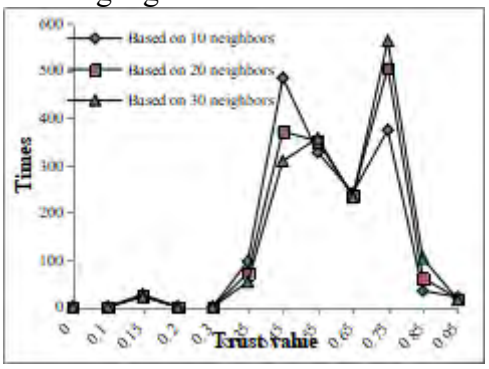

Fig. 4 Distribution of T_i

4) Experimental results and analysis: As shown in Fig.5, RMSE is changed with the number of neighbors, compare with ICF (Item-based Collaborative Filtering) and SVD-CF (Collaborative filtering algorithm based on SVD), when the number of neighbors less than 10, the RMSE of we proposed algorithm CFSVD-TF decreases exponentially; when the number of neighbors more than 10, the RMSE of we proposed algorithm CFSVD-TF changes tend to be stable; CFSVD-TF can do well at 10 neighbors and RMSE = 0.9762, the accuracy compared to SVD-CF is improved 0.53%; When neighbors from 12 to 20, RMSE has a few rises again, therefore the number of neighbors is an import factor for CFSVD-TF. SVD-CF and CFSVD-TF at RMSE are better than ICF. RMSE of CFSVD-TF always keeps declining at [0, 20], has a better performance than SVD-CF. That means the algorithm proposed in this paper is effective and feasible, not only effectively increase the data density, but improve the prediction accuracy.

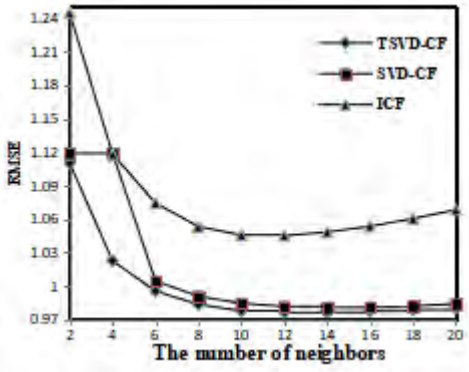

Fig. 5 RMSE under different recommendation strategies

5. Conclusions

In this paper, we propose CFSVD-TF algorithm based on analysis problems of traditional item-based collaborative filtering and sparse rating matrix. Firstly, SVD is used to alleviate the problem of data sparsity. Then trust factor between items is used to change the status of similarity between items as the sole determinant to prediction results. In the experiments presented, we show that our proposed method has better prediction accuracy than SVD-CF and ICF, is enough to satisfy the application area. In future work, we will study SVD with feedback, and combine it with higher feature vector. According to the personalized information between items, further improve the recommendation accuracy.

References

[1]. Pan Y, Wu D, Olson D L. Online to offline (O2O) service recommendation method based on multi-dimensional similarity measurement[J]. Decision Support Systems, 2017, 8 (3):1-8.

[2]. Moradi P, Ahmadian S. A reliability-based recommendation method to improve trust-aware recommender systems [J]. Expert Systems with Applications, 2015, 42(21): 7386-7398.

[3]. Han Yanan, Cao Han, Liu Liangliang. Collaborative filtering recommendation algorithm based on score matrix filling and user interest [J]. Computer Engineering, 2016, 42 (1): 36-40.

[4]. Wang Q M, Liu X, Zhu R, et al. A New Personalized Recommendation Algorithm of Combining Content-based and Collaborative Filters[J]. Computer & Modernization, 2013, 1(8):64-67.

[5]. Kong Wei-liang. Research on the Key Problems of Collaborative Filtering Recommender System [D]. Central China Normal University, 2013.

[6]. Wang Q, Liu X, Zhang S, et al. A Novel APP Recommendation Method Based on SVD and Social Influence[C]// International Conference on Algorithms and Architectures for Parallel Processing. Springer International Publishing, 2015: 269-281.

[7]. Paterek A. Improving Regularized Singular Value Decomposition for Collaborative Filtering[C]// Kdd Cup and Workshop. 2007,39-42.

[8]. Tsai C F, Hung C. Cluster ensembles in collaborative recommendation [J]. Applied Soft Computing, 2012, 12(4): 1417-1425.

[9]. Chen Ting, Zhu Qing, Zhou Mengxi, et al. Trust-based recommendation algorithms in social networks [J].Journal of Software, 2017, 28 (3): 721-731.

[10]. Ghazanfar M A, Prugel-Bennett A. The Advantage of Careful Imputation Sources in Sparse Data-Environment of Recommender Systems: Generating Improved SVD-based Recommendations [J].Informatics, 2013, 37(1):61-92.

[11]. Zhou X, He J, Huang G, et al. SVD-based incremental approaches for recommender systems[J]. Journal of Computer & System Sciences, 2015, 81(4):717-733.

[12]. YUAN Zhengwu, CHEN Ran. Collaborative filtering recommendation algorithm based on multi-level hybrid similarity[J]. Journal of Computer Applications, 2018, 38(3): 633-638.