A New Time-Aware Collaborative Filtering Intelligent Recommendation System

Weijin Jiang[1, 2, 3], Jiahui Chen[1, *], Yirong Jiang[4, *], Yuhui Xu[1], Yang Wang[1], Lina Tan[1] and Guo Liang[5]

Abstract: Aiming at the problem that the traditional collaborative filtering recommendation algorithm does not fully consider the influence of correlation between projects on recommendation accuracy, this paper introduces project attribute fuzzy matrix, measures the project relevance through fuzzy clustering method, and classifies all project attributes. Then, the weight of the project relevance is introduced in the user similarity calculation, so that the nearest neighbor search is more accurate. In the prediction scoring section, considering the change of user interest with time, it is proposed to use the time weighting function to improve the influence of the time effect of the evaluation, so that the newer evaluation information in the system has a relatively large weight. The experimental results show that the improved algorithm improves the recommendation accuracy and improves the recommendation quality.

Keywords: Fuzzy clustering, time weight, attenuation function, Collaborative filtering method, recommendation algorithm.

1 Introduction

The problem of data overload on the Internet platform makes it more and more difficult for users to extract real and useful information. Therefore, the intelligent recommendation system emerges as the times require. Taking the commodity selection of the e-commerce platform as an example, the intelligent recommendation system can filter out the user's most interested items by analyzing the user's interest characteristics and purchase behaviors. Also, it can improve the information matching efficiency in the massive data, and the user experience [Chen, Teng and Chang (2015)].The existing

[1] Key Laboratory of Hunan Province for New Retail Virtual Reality Technology, Hunan University of Technology and Business, Changsha, 410205, China.

[2] Institute of Big Data and Internet Innovation, Hunan University of Technology and Business, Changsha, 410205, China.

[3] School of Computer Science and Technology, Wuhan University of Technology, Wuhan, 430073, China.

[4] Tonghua Normal University, Tonghua, 134002, China.

[5] School of Bioinformatics, University of Minnesota, Twin Cities, USA.

[*] Corresponding Authors: Chen Jiahui. Email: cjh810663304@163.com;
Yirong Jiang. Email: lxh_yyy@163.com.

CMC. doi:10.32604/cmc.2019.05932

recommendation algorithms mainly include using user behavior data, user and item feature information, time and location, other context information, and social network data for recommendation.

The collaborative filtering recommendation algorithm achieves by obtaining the user's preference information, calculating the similarity between users (or between projects) and predicting the target user's rating of the target project based on similarity. The key step of collaborative filtering recommendation algorithm is to calculate the similarity between users (or between projects) [Bobadilla, Ortega and Hernando (2013)]. Scholars at home and abroad have carried out a series of studies on similarity measures in collaborative filtering algorithms. Zhou et al. [Zhou, He and Huang et al. (2015)] proposed an incremental method based on SVD to calculate the singular value decomposition of the original matrix step by step to solve the sparse problem and meet the changing needs of the dynamic user interest. Ramezani et al. [Ramezani, Moradi and Akhlaghian (2014)] used UTAOS to handle sparse and high-dimensional matrix problems and used subspace clustering to construct neighbor user trees. In addition, the user preferences among similar users can be found by uninterested items, rather than just comparing interesting items. In the similarity-based recommendation, Koohi et al. [Koohi and Kiani (2016)]. studied the optimization of neighbor number selection, compared the different methods to consider different numbers of neighbors, and proposed a neighbor-user clustering collaborative filtering method based on subspace clustering. Bobadilla et al. [Bobadilla, Hemando, Orteqa et al. (2012)] proposed combining the Mean Square Difference (MSD) and Jaccard coefficients to form a Jaccard-based Mean Square Difference (JMSD). When the above method encounters a cold start problem (i.e., when the rating information of new users and new items is small), its accuracy is affected. Sun et al. [Sun, Wu, Liu et al. (2013)] proposed the use of binary network technology to improve the collaborative filtering algorithm. The main idea is to introduce a bipartite network to describe the recommendation system in the collaborative filtering algorithm, and to use grey correlation to measure user similarity and project similarity. Liu et al. [Liu, Qu, Li et al. (2010)] use the idea of clustering to search for user neighbors. The core idea is to add the edge with the largest weight in the overlay network to the nearest neighbors in turn, classify the nodes that have loops in the set, and then use the unsupervised learning method K-means algorithm. For the second clustering, this method can improve the accuracy of the user's similarity, but it is not easy to update dynamically. Reference [Jojic, Shukla and Bhosarekar (2011)] proposed a quality-aware Web service recommendation method based on factorization machine. Reference [Wang, Yu, Feng et al. (2014)] proposed a collaborative filtering recommendation algorithm based on time series behavior. Choi et al. [Choi, Ko and Han (2012)] studied the parallel collaborative filtering recommendation algorithm based on graph walk. Wang et al. [Wang, Ma, Cheng et al. (2016)] studied the collaborative filtering recommendation algorithm that integrates the characteristics of social networks. Guo et al. [Guo, Ma, Chen et al. (2012)] studied the recommendation method of tourism group with fusion collaborative filtering and user preference. Wu et al. [Wu, Chen, Liu et al. (2012)] studied the recommendation algorithm for integrating user social status and matrix decomposition. Liu et al. [Liu, Chen, Xiong et al. (2012)] studied the collaborative filtering algorithm based on user feature optimization clustering. Ren et al. [Ren, Zhu et al. (2013)] studied a multi-feature fusion software developer recommendation method. Reference [Yang, Steck, Guo et al. (2012)] studied an efficient social network friend recommendation program. Liu et al. [Liu, Xiang, Chen et al. (2012)] studied a superimposed joint clustering recommendation model based on social networks. Yang et al. [Yang, Steck, Guo et al. (2012)] studied the personalized recommendation mechanism based on collaborative filtering in cloud computing environment. Leng et al. [Leng, Lu and Liang (2014)] studied stack noise reduction. Self-encoder tag collaborative filtering recommendation algorithm. Leng et al. [Leng, Liang, Ding et al. (2013)] studied the online recommendation of online social network users based on cross-platform research. Huo et al. [Huo, Zheng and Gao (2018)] studied the microblog friend recommendation method based on social circle discovery and user trust degree communication.

None of the above algorithms considers the impact of project relevance and user interest attenuation on the recommendation accuracy. Based on this, this paper proposes the following solutions. First, the project attribute fuzzy matrix is introduced. The project correlation is measured by the fuzzy clustering method. All project attributes are classified, and the weight of the project relevance is introduced in the user similarity calculation, so that the nearest neighbor search is more accurate. Second, in the prediction scoring section, taking into account the user's interest decay with time, it is proposed to use the time weighting function to improve the impact of the time effect of the evaluation, so that the newer evaluation information in the system has a relatively large weight.

2 Improved collaborative filtering algorithm

2.1 Project relevance

In the traditional collaborative filtering-based recommendation algorithm, the specific item information is often not considered [Shi, Xia and Liu (2018); Liang, Li, Zhang et al. (2018)]. The user-item evaluation matrix is mainly used to calculate the similarity of the user. Due to the subjective nature of this matrix, it often fails to reflect the real relationship between the projects, especially the project properties. Signs can often be assigned to multiple sub-attributes. Items with the same or similar sub-attributes are more recommendable when solving nearest neighbors [Chen, Gu and Chang (2018); Liu, Jing and Yu (2018)]. Therefore, this article considers the analysis of the characteristic attributes of the project itself, measures the project relevance through the fuzzy clustering method, and classifies all project attributes [Guo, Wang and Hou (2018)]. In this paper, the project attribute fuzzy matrix is introduced. As shown in Tab. 1, multiple attributes corresponding to each project are distinguished by relevance. This paper divides the project property fuzzification into four fuzzy collections: Highly correlated (Q1), General related (Q2) Weakly related (Q 3) and irrelevant (Q4).

The relevance between projects is very critical for obtaining a user's similar set of nearest neighbors. Generally, similar users tend to have a more consistent view of certain specific projects. For example, cigarettes, lighters and other items, to a certain extent, skirts, lipstick and other items, there are more obvious differences in the properties. The users who like the former and those who like the latter also have more obvious group differences. So, for the number of items, a large number of scoring matrices can be classified according to the nature of the attributes. When calculating the similarity of the user, if the correlation between the items can be measured, the nearest neighbor user's solution will be more accurate [Meng, Rong, Tian et al. (2018)]. In order to calculate the correlation between two items, the article first fuzzy the project attributes into different fuzzy sets, and then calculate the local fuzzy distance between different features. Here, the measurement is performed using the Euclid distance.

Table 1: The fuzzy matrix of item attributes

Projects	Attributes			
	Attribute 1	Attribute 2	…	Attribute m
Project 1	Highly Correlation	Weak Correlation	…	Strong Correlation
Project 2	Weak Correlation	General Correlation	…	General Correlation
⋮	⋮	⋮	⋮	⋮
Project n	Highly Correlation	Weak Correlation	…	Irrelevant

$$d_{ij} = \frac{1}{m}\sum_{k=1}^{m}\left|\left(a_{i,k} - a_{j,k}\right)^{2}\right|^{\frac{1}{2}} \tag{1}$$

Among them, $a_{i,k}$ is the membership value of the kth attribute of item i. The fuzzy clustering analysis method is used to solve the similarity between different attributes to construct a fuzzy matrix:

$$X = E - \begin{bmatrix} 0 & d_{12} & \cdots & d_{1n} \\ d_{21} & 0 & \cdots & d_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ d_{n1} & d_{n2} & \cdots & 0 \end{bmatrix} - \begin{bmatrix} 1 & x_{12} & \cdots & x_{1n} \\ x_{21} & 1 & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & 1 \end{bmatrix} \tag{2}$$

Where $x_{i,j}$=1 – d

For the properties of any fuzzy similarity matrix, there must be a minimal natural number k (k<n), such that the transitive closure t (X)=Xk. For all natural numbers greater than k, there is always Xl=Xk. Therefore, we can use a quadratic method to solve the transitive closure of R, that is t(X)=X^{*}

Further, by classifying the fuzzy λ-cutting matrix of the similarity matrix Xλ, the classification of Xλ at the λ level can be obtained. For different values of the condition $\lambda \in [0, 1]$, the final classification is often different. In order to more accurately solve the correlation between items, the F-statistic is used to determine the optimal value of λ .Then Project properties are categorized.

First calculate the average of the membership values of the attribute j, which are:

$$\overline{p_j} = \frac{d_{1j} + d_{2j} + \ldots + d_{nj}}{n} = \frac{\sum_{i=1}^{n} d_{ij}}{n} \tag{3}$$

Then, for m attribute indexes, $\overline{p}$ = ($\overline{p}$1, $\overline{p}$2, , $\overline{p}$m) can be used to represent the center vector of the relevance of each item attribute.Since the number of items to be classified depends on the value of λ, we assume that the number of item classifications is t, and the number of items in the k-th class is Attrk. There are: $\sum_{t=1}^{t} \mathrm{Attr}_k$. Let the mean value of each attribute relevance in the k-th class be denoted by $q_1^{(k)}$, $q_2^{(k)}$, and $q_3^{(k)}$, respectively.

Then the cluster center vector of the k-th class is $\overline{q}^{(k)} = (\overline{q}_1^{(k)}, \overline{q}_2^{(k)}, \ldots, \overline{q}_m^{(k)})$, where

$\overline{q}_j^{(k)} = \sum_{t=1}^{\mathrm{Attr}_k} \frac{d_{ij}^{(k)}}{\mathrm{Attr}_k}, (j = 1,2, \ldots, m)$.

Then calculate the Euclidean distance between the cluster center vector a and the center vector p of each class.

$$\overline{\rho_k} = \left| \sum_{i=1}^{m} \left(\overline{q_i}^{(k)} - \overline{p_i} \right)^2 \right|^{\frac{1}{2}} \tag{4}$$

Next, the distance between the item attribute relevance in each class and the cluster center vector of the class is compared.

$$\rho_i^k = \left| \sum_{\theta=1}^{m} \left(d_{i\theta}^{(k)} - \overline{q}_\theta^{(k)} \right)^2 \right|^{\frac{1}{2}} \tag{5}$$

Then you can get the sum of the distance of all the project attributes in the class and the cluster center vector of the class.

$$\rho_{sum} = \sum_{k=1}^{t} \sum_{i=1}^{\mathrm{Attr}_k} \left| \sum_{\theta=1}^{m} \left(p_{i\theta}^{(k)} - \overline{p}_\theta^{(k)} \right)^2 \right|^{\frac{1}{2}} \tag{6}$$

Finally, we can find the F statistic:

$$F = \frac{\sum_{k=1}^{t} Attr_k \left| \sum_{i=1}^{m} \left(\overline{q_i}^{(k)} - \overline{p_i} \right)^2 \right|^{\frac{1}{2}} / (t-1)}{\sum_{k=1}^{t} \sum_{i=1}^{Attr_k} \left| \sum_{\theta=1}^{m} \left(\overline{q}_{i\theta}^{(k)} - \overline{q}_\theta^{(k)} \right)^2 \right|^{\frac{1}{2}} / (n-t)} \tag{7}$$

Since the F statistic follows the F-distribution of degree of freedom t-1, n-t, we can see from the expression that the larger the value of F, the more obvious the gap between categories and the more reasonable the classification. Obviously, for the similarities between users based on item classification proposed in this paper, the accuracy of nearest neighbor matching is also higher.

After completing the above clustering, all project attributes are divided into several categories. Thus, we can set the similarity of items belonging to the same class to 1; otherwise, it means 0:

$$\mathrm{RS}\left(I_i, I_j\right)=\begin{cases}1, \textit{if } I_i \textit{ and } I_j \textit{ belong to some category} \\ \qquad 0, \textit{otherwise}\end{cases} \tag{8}$$

2.2 Attenuation function

In the collaborative filtering recommendation model, user evaluation tends to show attenuation characteristics over time. The influence of the previous evaluation on the current prediction value is inversely proportional to the length of the time span, that is, the long-term evaluation information in the system is time-efficient. Low-level information, in the process of making recommendations based on the user's interests, is less recommended than the fresh evaluation information [Tang, Zhang and Yang (2018)]. In order to fully reflect the influence of the "time effect" on the recommendation results, this paper adds a decay function to the prediction scoring process. This gives the user a relatively larger weight for the recently tagged item than the earlier tagged item, making the new evaluation more useful. The time weighting function is defined as follows:

$$\mathrm{f}\left(\mathrm{u}, \mathrm{v}, I_k\right)=\exp\left\{-\left(1+\delta\left|t\left(u, I_k\right)-t\left(v, I_k\right)\right|\right)\right\} \tag{9}$$

Among them, $\mathrm{t}(\mathrm{u}, \mathrm{I_k})$ and $\mathrm{t}(\mathrm{v}, \mathrm{I_k})$ respectively represent the time when user u and user v have evaluated item $\mathrm{I_k}$, and δ is the time decay parameter.

2.3 Modified user similarity calculation

According to the classification of the project attributes, the weight of the project relevance is introduced in the user similarity calculation, so that the neighbor users found are more accurate. This method is a combination of user-based and project-based collaborative filtering algorithms, but it is different from the previous way of linear combination of the two, but the two are combined in a non-linear way. That means in the process of computing user similarity, the weight of project relevance is introduced, that is, integrating projects based on users. At the same time, the relevance of the project is not calculated using the Pearson correlation coefficient, but is calculated based on the project feature attributes, which is more objective. The improved user similarity formula is as follows:

$$\mathrm{sim}\left(U_a, U_b\right)=\frac{\sum_{i\in I_{ab}}\left(r_{a,i}-\overline{r_a}\right)\left(r_{b,i}-\overline{r_b}\right)\mathrm{RS}\left(I_i, I_{\mathrm{e}}\right)}{\sum_{i\in I_{ab}}\left(r_{a,i}-\overline{r_a}\right)\left(r_{b,i}-\overline{r_b}\right)\mathrm{RS}\left(I_i, I_{\mathrm{e}}\right)} \tag{10}$$

Among them, I_e is the score item to be predicted, $\mathrm{sim}(U_a, U_b)$ is the similarity of users U_a and U_b based on I_e, and $\mathrm{rel}(I_i, I_e)$ is the relevance of items $I_i$and I_e. In the process of computing user similarity, the project relevance $\mathrm{rel}(I_i, I_e)$ is added, which reduces the interference of the low relevance project on the user similarity calculation and makes the selection of the nearest neighbor user more accurate.

2.4 Predictive score

In order to achieve more accurate real-time recommendations, in the prediction score, we take the time factor into account and improve the weighted prediction score of the unrated item I_e by the target user u to $P(u, I_e)$:

$$\mathrm{P}(u,I_e)=\overline{r_u}+\frac{\sum_{v\in k}\mathrm{sim}(\mathrm{u},\mathrm{v})\left(r_{v,I_e}-\overline{r_v}\right)\mathrm{f}(\mathrm{u},\mathrm{v},I_e)}{\sum_{v\in k}\mathrm{sim}(\mathrm{u},\mathrm{v})\mathrm{f}(\mathrm{u},\mathrm{v},I_e)} \tag{11}$$

wherein, it represents the average score of $\overline{r_u}$ user, and K is the set of neighbor users of the target user u. $\mathrm{sim}(u, v)$is the similarity value between user u and user v in the nearest neighbor set. r_{v,I_e} is the user v rating of the item I_e.

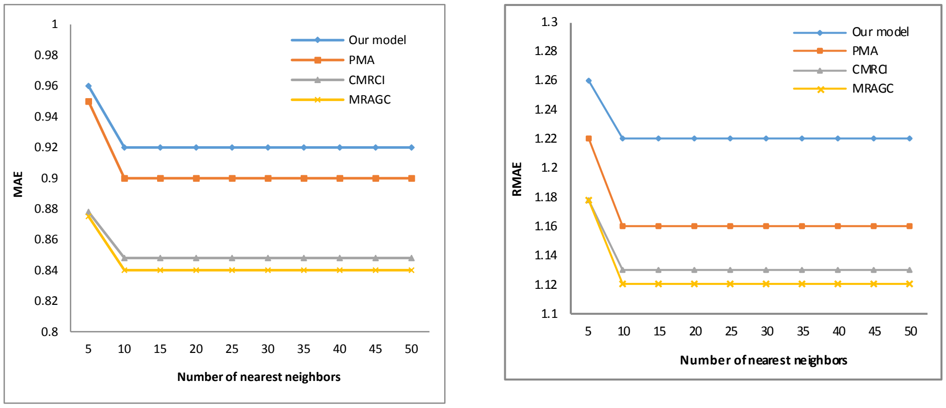
3 Experimental results and analysis

Experimental data uses a user-rated movie data set ml-data provided by the Movie Lens website. This data set contains 943 users’ 100,000 ratings data for 1682 movies. The rating value is an integer from 1 to 5, and the user’s preference of a movie is proportional to the size of the number. The text experiment randomly selected 400 users' rating data for 1500 movies. In addition, each movie was described by 20 attribute items, namely: unknown, Action, Adventure, Animation, Children’s, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western, etc. The project attribute fuzzy matrix is constructed according to the above attributes to classify the movies in the sample set.

This article compares the performance of the algorithm from two aspects, the accuracy of prediction and the accuracy of classification. The prediction accuracy rate is divided into Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [Yang, Steck, Guo et al. (2012)]. The smaller the MAE and RMSE values, the higher the accuracy of the prediction. The accuracy of the prediction reflects the accuracy of the algorithm for the prediction of unscored items. The difference in the number of nearest neighbors will result in a certain difference in the prediction accuracy. The nearest neighbor number K is used as the independent variable, and the attenuation factor δ is 0.5. The analysis is based on different The Collaborative Filtering Algorithm of Similarity Metrics.

Fig. 1. shows the algorithms proposed in this paper (our model) and various similarity measures based on the MovieLens dataset with different neighbor numbers (e.g., PMA [Ren, Zhu, Li et al. (2013)], CMRCI [Wu, Chen, Liu et al. (2012)], MRAGC [Tang, Zang and Yang (2018)]). The collaborative filtering recommendation accuracy comparison. In Fig. 2., the values of MAE and RMSE decrease as K increases. It can be seen that the MAE value of our model is lower than the collaborative filtering recommended MAE values of other measurement methods. At K=10, the MAE value of our model is 8.69% lower than that of PMA, 6.67%, 7 %, 1.18%. At K=10, the RMSE of our model is 8.19%, 5.88%, 1.75% is lower than PMA, CMRCI, and MRAGC, respectively. The algorithm proposed in this paper introduces the project relevance measurement method when users calculate the nearest neighbor [Xu, Zheng and Lyu (2016)]. We can see that the matching of the nearest neighbor user is more accurate. In addition, in the predictive scoring process, taking into account the user's interest decay over time, the use of the time-weighted function to improve the impact of the time effect of the evaluation, the relatively new weight of the newer evaluation information in the system, which also has an impact on the accuracy of the recommendation.

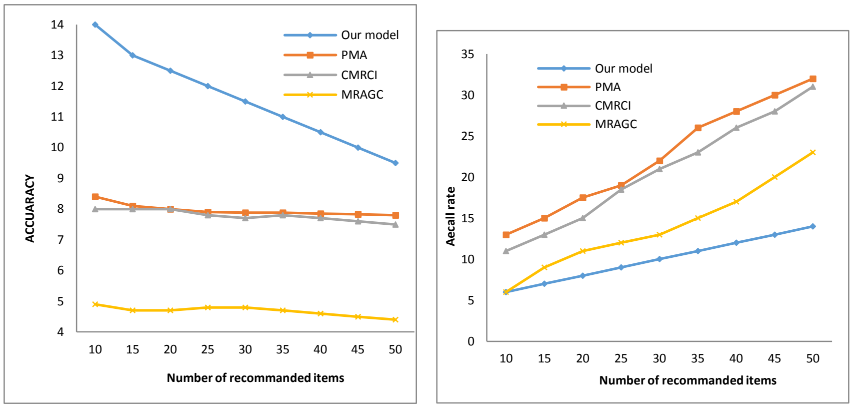
Fig. 2 shows the accuracy and recall of different recommended items under the MovieLens dataset. As can be seen from Fig. 2, our model can get the best recommended classification accuracy, and the effect is very obvious compared with other methods. In addition, it can be seen that the accuracy rate will decrease as the number of recommendations increases. Compared with PMA, when N=10, the accuracy rate of our model MODEL increases by 65.23%. the recall rate will increase with the increase in the number of recommendations. Compared with CMRCI, when N = 10, the recall rate of our model is increased by 53.42%.



a) MAE

b) RMSE

Figure 1: MAE and RMSE values of different similarity measures



a) ACCURACY

b) RECALL RATE

Figure 2: Precision and recall values of different recommendation items

4 Conclusion

This paper first analyzes the traditional collaborative filtering recommendation algorithm because it does not fully consider the influence of the correlation between projects on the recommendation accuracy. Also, it introduces the project attribute fuzzy matrix, measures the project relevance through the fuzzy clustering method, and classifies all the project attributes. Then, in the calculation of user similarity, the weight of the project relevance is introduced to make the nearest neighbor search more accurate. In addition, in the predictive scoring process, the time-weighted function is used to improve the time effect of the evaluation, taking into account the user's interest decay over time. The impact of the system makes relatively new evaluation information in the system relatively heavy. Experimental results show that the improved algorithm improves the recommendation accuracy and improves the recommendation quality.

Acknowledgement: This work was supported by the National Natural Science Foundation of China (61772196; 61472136), the Hunan Provincial Focus Social Science Fund (2016ZDB006), Hunan Provincial Social Science Achievement Review Committee results appraisal identification project (Xiang social assessment 2016JD05), Key Project of Hunan Provincial Social Science Achievement Review Committee (XSP 19ZD1005). The authors gratefully acknowledge the financial support provided by the Key Laboratory of Hunan Province for New Retail Virtual Reality Technology (2017TP1026).

References

Bobadilla, J.; Hemando, A.; Orteqa, F.; Gutiérrez, A. (2012): Collaborative filtering based on significances. *Information Sciences*, vol. 185, no. 5, pp. 1-17.

Bobadilla, J.; Ortega, F.; Hernando, A. (2013): Recommender systems survey.

Knowledge-Based Systems, vol. 46, no. 3, pp. 109-132.

Chen, M. H.; Teng, C. H.; Chang, P. C. (2015): Applying artificial immune systems to collaborative filtering for movie recommendation. *Advanced Engineering Informatics*, vol. 29, no. 4, pp. 830-839.

Choi, S. M.; Ko, S. K.; Han, Y. S. (2012): A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications*, vol. 39, no. 9, pp. 8079-8085.

Chen, J. T.; Gu, T. L.; Chang, L. (2018): A tourist group recommendation method combining collaborative filtering and user preferences.

http://kns.cnki.net/kcms/detail/23.1538.TP.20180416.1325.008.html

Guo, L.; Ma, J.; Chen, Z. M.; Jiang, H. R. (2012): Learning to recommend with social relation ensemble. *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 2599-2602.

Guo, N. N.; Wang, B. L.; Hou, Y. H. (2018)**:** Collaborative filtering recommendation algorithm based on characteristics of social network. *Journal of Frontiers of Computer Science and Technology*, vol. 12, no. 2, pp. 208-217.

Huo, H.; Zheng, D. Y.; Gao, L. P. (2018): Collaborative filtering with labels based on stack denoising auto encoder. *Journal of Chinese Computer Systems*, vol. 39, no. 1, pp. 7-11.

Jojic, O.; Shukla, M.; Bhosarekar, N. (2011): *A Probabilistic Definition of Item Similarity*. Chicago, USA.

Koohi, H.; Kiani, K. (2016): User based collaborative filtering using Fuzzy C-Means. *Measurement*, vol. 91, no. 3, pp. 134-139.

Liu, Z.; Qu, W.; Li, H.; Xie, C. (2010): A hybrid collaborative filtering recommendation mechanism for P2P networks. *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1409-1417.

Liu, Q.; Chen, E. H.; Xiong, H.; Ding, C. H. Q.; Chen, J. (2012): Enhancing collaborative filtering by user interests expansion via personalized ranking. *IEEE Transactions on Systems, Man and Cybernetics-B*, vol. 42, no. 1, pp. 218-233.

Liu, Q.; Xiang, B.; Chen, E. H.; Ge, Y.; Xiong, H. et al. (2012): Influential seed items recommendation. *Proceedings of the ACM Conference on Recommender Systems*, pp. 245-248.

Leng, Y. J.; Lu, Q.; Liang, C. Y. (2014): Survey of recommendation based on collaborative filtering. *Pattern Recognition and Artificial Intelligence*, vol. 27, no. 8, pp. 720-734.

Leng, Y. J.; Liang, C. Y.; Ding, Y.; Lu, Q. (2013): Method of neighborhood formation in collaborative filtering. *Pattern Recognition and Artificial Intelligence*, vol. 26, no. 10, pp. 968-974.

Liang, L. J.; Li, Y. G.; Zhang, N. N.; Zhang, X.; Wang, D. (2018): Recommendation algorithm combing user features and preference in optimized clustering.

Liu, H. F.; Jing, L. P.; Yu, J. (2018): Survey of matrix factorization based recommendation methods by integrating social information. *Ruan Jian Xue Bao/Journal* *of Software*, vol. 29, no. 2, pp. 340-362.

Hou, M. W.; Wei, R.; Wang, T. G.; Cheng, Y.; Qian, B. Y. (2018): Reliable medical recommendation based on privacy-preserving collaborative filtering. *Computers, Materials & Continua*, vol. 56, no. 1, pp. 137-149.

Ramezani, M.; Moradi, P.; Akhlaghian F. (2014): A pattern mining approach to enhance the accuracy of collaborative filtering insparse data domains. *Physica A: Statistical Mechanics and Its Applications*, vol. 408, no. 9, pp. 72-84.

Ren, Y. L.; Zhu, T. Q.; Li, G.; Zhou, W. L. (2013): Top-N recommendations by learning user preference dynamics. *Advances in Knowledge Discovery and Data Mining*.

Sun, G. F.; Wu, L.; Liu, Q.; Zhu, C.; Chen, E. H. (2013): Recommendations based on collaborative filtering by exploiting sequential behaviors. *Ruan Jian Xue Bao/Journal of Software*, vol. 24, no.11. pp. 2721-2733.

Shi, C. Q.; Xia, G. W.; Liu, J. P. (2018): Time weight increasing-based collaborative filtering algorithm. *Journal of Chinese Computer Systems*, vol. 39, no. 2, pp. 255-261.

Tang, M. D.; Zhang, T. T.; Yang, Y. T. (2018): QoS-awore web service recommendation based on factorization machine. *Chinese Journal of Computers*, vol. 41, no, 6. pp. 1080-1093.

Wu, L.; Chen, E. H.; Liu, Q.; Xu, L. L.; Bao, T. F. et al. (2012): Leveraging tagging for neighborhood-aware probabilistic matrix factorization. *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 1854-1858.

Wang, Z.; Yu, X.; Feng, N.; Wang, Z. (2014): An improved collaborative movie recommendation system using computation al intelligence. *Visual Languages and Computing*, vol. 25, no. 3, pp: 667-675.

Wang, S.; Ma, Y.; Cheng, B.; Yang, F.; Chang, R. (2016): Multi-dimensional QoS prediction for service recommendations. *IEEE Transaction on Services Computing*, pp. 1.

Xu. J.; Zheng, Z.; Lyu, M. R. (2016): Web service personalized quality of service prediction via reputation based matrix factorization. *IEEE Transactions on Reliability*, vol. 65, no. 1, pp. 28-37.

Yang, X. W.; Steck, H.; Liu, Y. (2012): Circle-Based recommendation in online social networks. *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1267-1275.

Yang, X. W.; Steck, H.; Guo, Y.; Liu, Y. (2012): On top-K recommendation using social networks. *Proceedings of the ACM Conference on Recommender Systems*, pp. 67-74.

Zhou, X.; He, J.; Huang, G. Y.; Zhang, Y. C. (2015): SVD-based incremental approaches for recommender systems. *Journal of Computer and System Sciences*, vol. 81, no. 4, pp. 717-733.