2021 the 6th International Conference on Control, Robotics and Cybernetics

Collaborative Filtering Recommendation Algorithm Based on Improved BiasSVD

2021 6th International Conference on Control, Robotics and Cybernetics (CRC) | 978-1-6654-2437-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CRC52766.2021.9620165

Jiaqin Fan
School of Electrical and Information Engineering
Wuhan Institute of Technology, WIT
Wuhan, China
1252411562@qq.com

Ruonan Ji
School of Electrical and Information Engineering
Wuhan Institute of Technology, WIT
Wuhan, China
jair0327@163.com

Yi Tian
School of Electrical and Information Engineering
Wuhan Institute of Technology, WIT
Wuhan, China
13407115931@163.com

***Abstract*—In the era of information explosion, the internet platform with massive information provides personalized services for users by using recommendation system. With users and projects growing ever more, the data sparsity problem may affect the cold start of projects. To solve the above problem, an improved collaborative filtering recommendation algorithm is proposed in this paper. Firstly, the biasedSVD algorithm is used to reduced dimension of user rating matrix, which brings the less complexity and alleviates the problem of data of sparsity. Secondly, the user's behaviors are analyzed and a novel weight named the time-preference weighting factor is proposed to describe the user's rating preference for items. Experimental results based on MovieLens data set show that the proposed method outperforms the state-of-the-atrs.**

Keywords—Collaborative filtering, Data sparsity, BiasSVD, Time-preference weighting factor

I. Introduction

Nowadays, collaborative filtering recommendation algorithm is the most widely used in all platforms, which greatly improves recommendation system [1]. The core idea of collaborative filtering is to provide recommendation according to the nearest neighbor set which is similar to the interest preference of the target user. Collaborative filtering recommendation algorithm exists some problems, such as data sparsity, cold start, scalability, poor recommendation quality and so on.

In this paper, an improved collaborative filtering recommendation algorithm based on BiasSVD is proposed, which uses KNN algorithm program to calculate the user's similarity and find the nearest neighbor set and decomposes the score matrix by utilizing SVD decomposition on unknown items to predict a relatively complete user rating matrix. Considering the user preferences in the project change along with time, a novel Time-preference weighting factor based on forgetting curve is proposed to predict the user's rating of the item, based on which the final personalized recommendation is made for the user.

II. Research Background

A. System Architecture

Based on the user rating matrix, the proposed method where BiasSVD [3] fused the time-preference weighting factor is used to calculate the prediction score. The whole framework of the proposed algorithm is shown in Fig. 1.

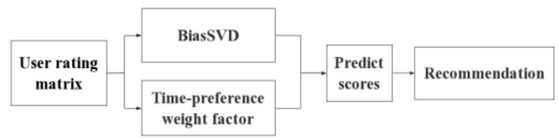

Fig. 1. Algorithm framework flow chart

B. Time-Preference Weighting Factor

According to the Ebbinghaus Forgetting Curve [4-6], the attenuation trend of user- preference curve is similar to Ebbinghaus memory curve. With passage of time, the attenuation rate of user's preference for items moves fast at first, then slows down and finally tends to be stable.

Fig. 2 shows the variation of the rating of users to the item with the rating time. The horizontal axis T represents the rating time of user, and Y represents the user's rating on the item. The origin t_0 is the reference time. The farther away from the origin, the longer is the time interval, and the smaller is the weight for reference.

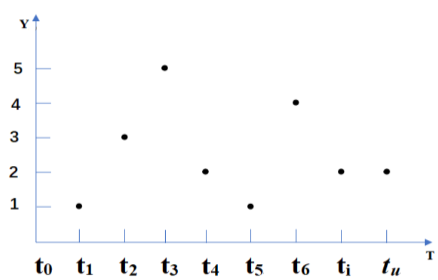

Fig. 2. Schematic diagram of user project rating

According to the above two-dimensional coordinate system and the Ebbinghaus forgetting curve, the improved user's time-preference weighting factor function $W(u,i)$ can be defined as below:

$$W(u,i) = \begin{cases} \dfrac{e^{-(t_c - t_i)}}{t_c}, t_c \neq t_i \\ 1, t_c = t_i \end{cases} \tag{1}$$

where t_c represents the current time. t_i represents the rating time of user on item "i". The function of time-preference weighting factor $W(u,i)$ decelerates with the increase of t_c, and then becomes stable gradually. The experimental $W(u,i)$ function is associated with the time interval of user's evaluation, and the reference weight ratio is different as evaluation time, which can reflect the user's preference accurately, so that the quality of recommendation is better.

C. SVD Decomposition

The user's rating data set is extremely sparse, which has a negative effect on the results of recommendation for the target users. The SVD decomposition model [7-8] is used to complete the user scoring matrix, and its disturbance to the user scoring matrix is small, which could reduce the prediction scoring errors as well as increase the quality of recommendation results.

Singular Value Decomposition (SVD) can project the high-dimensional matrix to lower subspace, so as to obtain the intrinsic structure of the matrix directly. The sparsity of data can be alleviated to a certain extent [9-11] as well.

Assuming A be a (m×n) matrix, denoted as $A_{m\times n}$. When m=n, it can be decomposed by traditional eigenvalue method. If m ≠ n, matrix A can be decomposed into, $A_{m\times n}=USV^T$. U is a (m×m) orthogonal matrix, V is an (n×n) orthogonal matrix, and S is a (m×n) matrix. S*, the first k-order matrix of S, is a positive semi-definite diagonal matrix, denoted as:

$$S^{*}=diag\ (a_1,\ a_2,\ \cdots a_k),\ k\ \delta\ \min\ (m,\ n)$$

Among them ($a_1, a_2, \cdots a_k$) are the singular values of matrix A. The largest *k* eigenvalues of the matrix are obtained by dimensionality reduction, and these k eigenvalues are used to reconstruct the matrix. The corresponding k columns and k rows in matrix u and matrix v are used to obtain reconstructed matrices u and v. Then the dimension of matrix $A_{k\times k}$ is reduced in term of the dimensionality and obtained by *U*, *S* and *V*, as follows:

$$A_k = U_k S_k V_k^T \tag{2}$$

D. Improved BiasSVD

BiasSVD [12-13] decomposes the user rating matrix into user correlation matrix and item correlation matrix through SVD to achieve the purpose of dimension reduction and reconstruction of the original scoring matrix. However, the scores predicted by the method of matrix decomposition have some deviations. Then, the user preferences, items biases, and the average overall score are added in the prediction formula to make the predicted score closer to the true score value. Namely, the prediction scoring formula for BiasSVD is as follows:

$$r'_{ui} = \mu + b_u + b_i + p_u^T \cdot q_i \tag{3}$$

where μ is the average of all scoring records in the training set, and represents the overall scoring situation in the training set, which is a fixed constant. b_u is expressed as user bias, which is independent of the factors of item characteristics and represents the user preference for himself and specific items. b_i is the bias of item *I*. It is independent of the user's interest and indicates the scoring situation of a particular item. For example, movies with higher scores on Douban tend to have higher overall user ratings.

The default option of BiasSVD algorithm is that the user bias is in a long-term stable state. However, in the actual environment, the user preference changes with time, resulting in the same state of user bias in different periods.

In this paper, time-preference weighting factor is introduced into user bias of BiasSVD [14-16], so that users' rating preference changes with time. Combining experimental results with the time-preference weighting factor, different evaluation time-reference weights are different, so the recommendation accuracy can be significantly improved. Combining formula (3) and introducing time-preference weighting factor, the improved calculation formula of user prediction score is defined as follows:

$$\overline{R}_{ui} = \mu + b_u \cdot W(u,i) + b_i + p_u^T \cdot q_i \tag{4}$$

III. The Process of System Implementation

A. Collecting User Rating Matrix

Based on the downloaded data set, data preprocessing was performed to construct the user rating matrix $R_{m\times n}$ as shown in Table 1. The user set is $U=\{U_1,U_2,\cdots U_i\cdots U_m\}$, where m represents the number of users; And the item set is represented by $I=\{I_1,I_2,\cdots I_i,\cdots I_n\}$, where *n* represents the number of items. The value of R_{ui} indicates the user's rating to the item, with a score of 1-5, and 0 indicates that the user has not rated the item. The user-item matrix is formed according to the number of users *m* and the number of items *n*, and the nearest number k is input.

TABLE I. User-Item Rating Matrix

Users	Items					
	I_1	I_2	...	I_m	...	I_n
U_1	1	1	...	0	...	4
U_2	0	2	...	5	...	1
...	0	1	...	4	...	0
U_m	1	5	...	2	...	1

B. Training the Model

Initializing the matrices P and Q matrices, $b_u=0, b_i=0$, the number of iteration steps is input. The step size of gradient descent is γ, and the penalty factor is λ. Perform BiasSVD model training on the training set to obtain optimized P and Q matrices, b_u and b_i vectors.

C. Calculating The Similarity

The similarity with the target users is calculated and placed in sequence, and the first N users are selected to generate the "nearest neighbor set" of the target *U*.

D. Predicting Score

The improved rating prediction calculation formula (4) is applied to predict the rating of target user to the item, based on the rating of the item “I” by the user “u” in the "nearest neighbor user set".

E. Recommended List

The top n items with the highest prediction scores are taken as the top-N recommendation set to the target users.

IV. Experimental Result and Analysis

A. Data Set Selection and Evaluation Standard

In order to verify the superiority of the improved algorithm, this paper selects the standard data set provided by the University of Minnesotahe. 100k MovieLens [17] data set is used. The evaluation indexes of the algorithm selected in the experiment are root mean square error (RMSE), mean absolute error (MAE), accuracy and recall rate. RMSE and MAE reflect the difference between the predicted score and the user's actual score. The formula to calculate the RMSE and MAE is as follows:

$$RMSE = \sqrt{\frac{\sum_{u,i\in T}(R_{ui}-\overline{R}_{ui})^2}{T}} \tag{5}$$

$$MAE = \frac{\sum_{u,i\in T}\left|R_{ui}-\overline{R}_{ui}\right|}{|T|} \tag{6}$$

$\overline{R_{ui}}$ represents the predicted scores, and R_{ui} represents the actual score. T represents the total number of items in the test set. The smaller are the RMSE and MAE values, the more similar is the predicted value to the real value, and the higher quality of recommendation is.

Accuracy and recall rate are used to measure the recommendation ability [18-20], where R(u) and T(u) represent the recommendation lists recommended to users based on the training set and the test set, respectively. The accuracy rate is the ratio of the user's favorite items to all recommended products in the recommended list based on the training set. The accuracy rate indicates the possibility that the user is interested in the recommended product. Formula for precision is as follows:

$$P = \frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|R(u)|} \tag{7}$$

The recall rate is defined as the ratio of the products favored by users in the recommendation list to the number of items in the test set, which means the probability of a product that users like is recommended. The formula is as follows:

$$R_E = \frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|T(u)|} \tag{8}$$

B. Analysis Experimental Result

The programming used in this experiment is Python [21-22]. In the 100k MOvieLens data set, the BiasSVD collaborative filtering recommendation algorithm is compared with the traditional recommendation algorithms UserCF and ItemCF. The result is shown in Fig. 3. It is clear that the accuracy of the BiasSVD algorithm is significantly higher than the others. The accuracy rate increases, when the proximity value is between 10 and 20. And it firstly decreases slowly and then increases in the proximity value with the range of 20-80. As the proximity value continues to increase, the accuracy rate tends to be stable, and finally stabilizes around the value of 2.67.

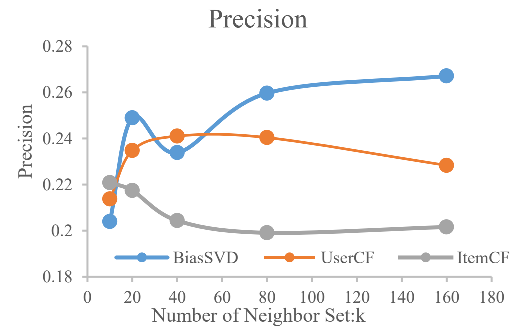

Fig. 3. Comparison of accuracy between different algorithms

It is shown in Fig. 4 that the recall rate of BiasSVD increases slowly when the value of k is between 10 and 20, and decreases slowly when the value is between 20 and 40. When the value is greater than 40, the recall rate increases slowly and finally stabilizes at about 0.14.

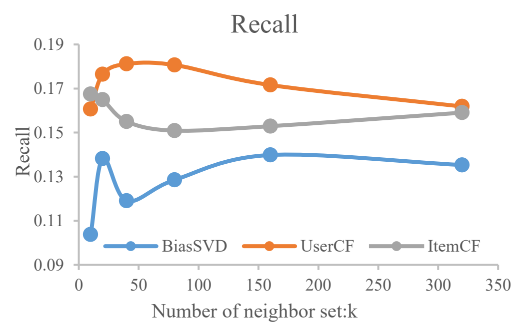

Fig. 4. The recall rate for BiasSVD, UserCF and ItemCF algorithm

As shown in Figure 5, in the MovieLens data set, the BaisSVD algorithm with time-preference weighting factor is generally lower than the traditional algorithm with the same k. The RMSE of the improved BiasSVD algorithm shows a downward trend when the adjacent value is between 5 and 10. When k=10, RMSE is at the lowest point. When the adjacent number is between 10 and 160, RMSE of the improved BaisSVD algorithm shows a small range of fluctuations, With the increase of the adjacent number, the RMSE finally stabilizes around 0.92.

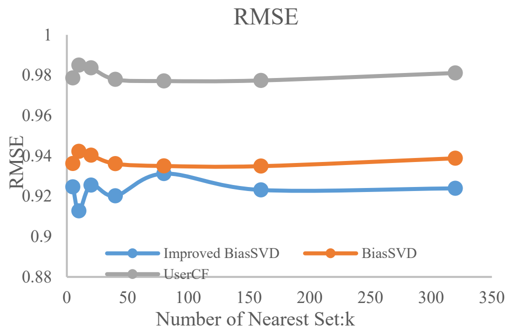

Fig. 5. Comparison of RMSE between improved collaborative filtering algorithm and traditional system filtering algorithm

As shown in Fig. 6, the MAE values of the traditional algorithms UserCF and BiasSVD are generally higher. When the proximity number is greater than 20, the Mae values of UserCF and BiasSVD show a small downward trend, and finally gradually stabilize at 0.77 and 0.74. The MAE of improved BiasSVD algorithm is always lower than UserCF and BiasSVD.

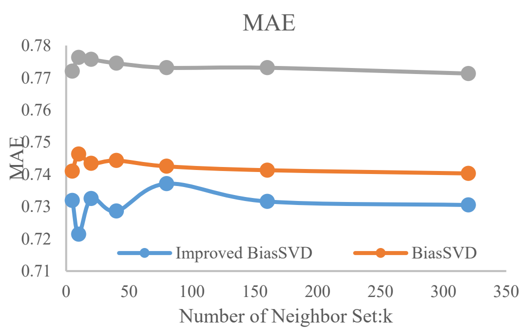

Fig. 6. The comparison between the improved collaborative filtering algorithm and the traditional system filtering algorithm in MAE

The above results show that the introducing of time-preference weighting factor based on user characteristics to the BiasSVD collaborative filtering algorithm can significantly reduce RMSE and MAE and improve the accuracy of recommendation compared with the traditional collaborative algorithm.

V. Conclusion

In this paper, the collaborative filtering recommendation algorithm based on BiasSVD was improved, and the time-preference weighting factor is incorporated. The experimental results showed that the retrieval complexity was reduced and the sparsity of data was alleviated. By introducing the time-preference weighting factor, the problem of constant user bias of BiasSVD was solved. The accuracy of prediction score was improved, and the algorithm performance was obviously improved.

Acknowledgment

I am very grateful to the people who gave me encouragement and help during this time.

References

[1] Wu C, Garg D, Bhandary U. Movie recommendation system using collaborative filtering[C]//IEEE 9th International Conference on Software Engineering and Service Science.

[2] Agrawal S, Jain P. An improved approach for movie recommendation system[C]//International Conference on IoT in Social, Mobile, Analytics and Cloud(I-SMAC), Palladam, 2017: 336-342.

[3] Li Jia, Zhang Mu. Research on Collaborative Filtering Algorithm Based on BiasSVD and Clustering User's Nearest Neighbors[J]. Modern Computer, 2020(13):30-34.

[4] Zhang Zhipeng, Zhang Yao, Ren Yonggong. Collaborative filtering recommendation algorithm based on time correlation and coverage weight[J]. Pattern Recognition and Artificial Intelligence, 2019, 32(04): 289-297.

[5] Li Yan, Ai Jun, Su Zhan. A collaborative filtering recommendation algorithm combining scoring time and user space[J]. Computer Applications and Software, 2018, 35(12): 247-252.

[6] Li Taoying, Zhang Xin, Chen Yan. Analysis of Vague Relations of Retail Products Based on Ebbinghaus Forgotten Curve[J]. Application Research of Computers, 2018,35(02): 462-465.

[7] Li Yao. An improved collaborative filtering recommendation algorithm [J]. Journal of Tianjin University of Technology, 2021, 37(01): 1-5.

[8] Niu R P, Liu G R, Li M. The inverse methods based on S-FEMs with an adaptive SVD regularization technique for solving Cauchy inverse heat transfer problems [J]. Engineering Analysis with Boundary Elements, 2019, 107: 79–95.

[9] Wang Chong. Research on collaborative filtering algorithm based on SVD and user clustering[D]. Qingdao: Qingdao Technological University, 2018.

[10] Wang Yan. Research and application of recommendation system based on SVD[D]. Taiyuan University of Technology, 2017.

[11] Liu Chao, Zhao Wenjing, Jia Yuzhen, Cai Guanyu. Collaborative filtering hybrid recommendation algorithm based on improved BiasSVD and clustering user nearest neighbors[J]. Computer Applications and Software, 2021, 38(05): 288-293.

[12] Wang Hong, Chen Gongping. Research on the Application of Implicit Semantic Model in Recommender System[J]. Journal of Guangdong Technical Normal University, 2020, 41(06): 9-13.

[13] Li Mingxiu, Wang Shujun, Jia Ru, Chen Lirong. Collaborative filtering algorithm based on fusion time information of multiple bias items[J]. Software Engineering, 2019, 22(06): 17-21+12.

[14] Zeng An, Gao Chengsi, Xu Xiaoqiang. A collaborative filtering algorithm combining time factors and user rating characteristics[J]. Computer Science, 2017, 44(09): 243-249.

[15] Bao Xuan, Chen Hongmei, Xiao Qing. Collaborative recommendation algorithm for points of interest integrated into time [J/OL]. Computer application, 2021-08-16:1-7.

[16] Xin Shuai. Research on hybrid recommendation algorithm based on time weight[D]. Liaoning University of Science and Technology, 2018.

[17] Wang Yonggui, Song Zhenzhen, Xiao Chenglong. Collaborative filtering recommendation algorithm based on improved clustering and matrix decomposition. Computer Applications, 2018, 38(4): 1001-1006.

[18] Tian Yu, Zhang Xiaochen, Wang Huabin. Information technology education test question recommendation algorithm integrating chapter information[J]. Journal of Anhui University (Natural Science Edition), 2020, 44(05): 46-55.

[19] Xiao Wenqiang, Yao Shijun, Wu Shanming. An improved top-N collaborative filtering recommendation algorithm. Computer Application Research, 2018, 35(1): 105-108, 112.

[20] Wei Hao, Zhang Wei, Guo Xinming. Collaborative filtering recommendation algorithm using cluster analysis[J]. Fujian Computer, 2021, 37(05): 1-4.

[21] Huang Yongchang. Scikit-learn machine learning common algorithm principle and programming practice. Beijing: Machinery Industry Press, 2018.

[22] Ivan Idris. Python data analysis. Beijing: People's Posts and Telecommunications Press, 2016.