Improved hybrid recommendation algorithm based on joint interpolation

QiChen Su¹*, Yang Su² and TianYi Liu²

¹ School of Mathematics and Statistics, Huizhou University, Huizhou, Guangdong, 516000, China
² School of Computer Science and Technology, Dongguan University of Technology, Dongguan, Guangdong, 523000, China

*Corresponding author’s e-mail: sqc20202001@hzu.ecu.cn

Abstract. Aiming at the sparsity problem of the underlying score matrix of the collaborative filtering algorithm, to solve the problem of poor filling effect when the existing filling method has a large difference in the scores of items between neighbors, an improved hybrid recommendation algorithm based on joint interpolation is proposed. The algorithm first uses joint interpolation to fill in the user’s rating matrix, and then uses the similarity between the filled data and the user and item information to predict the user’s rating of the item, and then compares the item’s rating with the user’s scores of similar items, impose penalties on scores that are far apart, and finally recommend the penalized scores from high to low. Experimental results show that the algorithm has a better recommendation effect.

1. Introduction

With the rapid development of the Internet, the amount of information has exploded. But this kind of information has a high level of redundancy, makes it difficult for people to find the information they really need quickly and effectively. Therefore, the recommendation system came into being. As one of the main means to solve the information overload, the recommendation system has received extensive attention in academia and industry. Many domestic and foreign scholars and large Internet companies have participated in the theoretical research and application practice of such systems [1-5].

Mainstream recommendation systems are divided into three categories: collaborative filtering-based, content-based, and hybrid recommendation systems [6-7]. Among them, collaborative filtering recommendation systems are the most widely used [8]. Its core idea is that similar users’ ratings of new items should be similar, and the same user’s ratings of similar items should be similar too. Based on the above two ideas, personalized recommendations can be made to users. So, the key of collaborative filtering recommendation systems is to find similar users for the target user, or find similar items that are rated by the target user. The core problem faced by the collaborative filtering recommendation system is the sparseness of the underlying score matrix [6], because the items that users have evaluated are only a small part of the huge number of item libraries.

In order to solve the above problems, [9] proposed a collaborative filtering recommendation algorithm based on item score prediction. It first calculates the similarity between items using the traditional similarity measurement method, then calculates the target user’s predicted score for the items, performs interpolation, and finally calculates the target user’s final score on the item based on the similarity between users; [10] proposed a collaborative filtering algorithm that combines the filling
method and the improved similarity measurement. When interpolation, the item similarity weight is introduced, and then the user’s predicted score is calculated using the new similarity, and finally a recommendation is generated; [11] proposed a recommendation algorithm that combines improved filling method and multi-weight similarity. On the basis of [10], the item score, popularity, item attributes and time factors are considered when calculating the interpolation, and then the multi-weight similarity is calculated, finally the target user’s final score of the item is calculated. The algorithm proposed in [9] only uses the user’s rating of items when calculating interpolation, ignoring the impact of item characteristics. [10-11] has made improvements on the basis of [9], adding the weight of item similarity when calculating interpolation, but when the scores of items between neighbors vary too much, the accuracy of the scores predicted by these methods is low [12].

In this paper, we proposed an improved hybrid recommendation algorithm based on joint interpolation. First, the joint interpolation method is used to interpolate the score matrix of the target user, since this algorithm can maintain good accuracy when the scores of items between neighbors vary too much [12]. Then improve the similarity calculation method in [11], use the new improved method to predict the target user’s score, and compare the predicted score with the already known user’s score of similar items, impose penalties on scores with large differences. Finally get the predicted score of the target user, and make recommendations according to the predicted score from high to low.

2. Improved hybrid recommendation algorithm

2.1. Joint interpolation
[13] uses equation (1) to calculate the interpolation of target user \( u \) to item \( j \):

\[
\hat{s}_{uj} = \mu_u + \sum_{v \in Pu(j)} \omega_{uv}^\text{user} (r_{vj} - \mu_v)
\]  

(1)

where \( \mu_u \) and \( \mu_v \) represents the average score of user \( u \) and user \( v \) respectively, \( r_{vj} \) represents the score of user \( v \) on item \( j \), \( Pu(j) \) is the \( k \)-th users who are most similar to the target user \( u \) and have commented on the item \( j \), \( \omega_{uv}^\text{user} \) represents the similarity between user \( u \) and user \( v \), which can be expressed by Pearson correlation coefficient or cosine similarity. In this paper, Pearson correlation coefficient was used.

2.2. Improved similarity measurement
A new similarity method proposed in [10] believes that when the features of the items are exactly the same, the second similarity is 1, otherwise the second similarity is 0, but there are actually some cases where part of the features of different items are the same. The second similarity calculated by the method in [10] is too one-sided. We divide the item features into structured features and unstructured features, then calculate the second similarity: if the item feature is a structured feature, then use the traditional cosine similarity method; if the item feature is an unstructured feature, the word frequency statistics method is used to calculate the similarity. First, calculate the item’s \( TF-IDF \) (term frequency-inverse document frequency) as shown in equation (2),

\[
TF-IDF(t_k, d_j) = TF(t_k, d_j) \log \frac{N}{n_k},
\]  

(2)

where \( TF(t_k, d_j) \) is the number of times the \( k \)-th keyword appears in the description of the \( j \)-th item, \( n_k \) represents the number of items containing the \( k \)-th keyword, and \( N \) represents the number of items. Then calculate the normalized weight, as shown in equation (3),
2.3. Users’ final score for items
We calculate the users’ final score for items in two steps. In the first step, we adopt the golden ratio idea [10], that is the users’ score is obtained by the weighted summation of the two similarity scores according to the golden ratio. This whole idea can be formulized by equation (5),

\[ P_{ui} = 0.618 \text{sim}_{1ui} + 0.312 \text{sim}_{2yz}, \]

with \( \text{sim}_{1ui} = \frac{\sum_{v \in S \cap Nsim(u, v)} r_{uv}}{\sum_{v \in S \cap Nsim(u, v)}} \) and \( \text{sim}_{2yz} = \frac{\sum_{i \in T \cap Nsim2(u, i)} r_{ui}}{\sum_{i \in T \cap Nsim2(u, i)}} \), where \( r_{ui} \) represents the rating of user \( u \) on item \( i \), \( S \) is the neighbor set of user \( u \), and \( T \) represents the neighbor set of item \( i \). Then, in the second step, we compare the calculated predicted score \( P_{ui} \) with the target user \( u \)'s known score for similar item \( s \), if the absolute value of the difference between the two is greater than \( \beta \), a penalty is imposed to get the final user \( u \)'s score for item \( i \), see equation (6),

\[ \hat{P}_{ui} = P_{ui} - \lambda (P_{ui} - r_{us}). \]

The framework of the collaborative filtering algorithm based on joint interpolation we proposed can be summarized as follows:
1. Use the joint interpolation method to calculate the interpolation of the target user’s score matrix;
2. Calculate the first similarity \( Nsim(u, v) \) of the item and extract the item feature. If the item feature is a structured feature (such as the attribute contains a value), use the traditional cosine similarity method to calculate the second similarity \( Nsim2(u, v) \); If the item feature is an unstructured feature (such as a text description), use equation (2), (3), (4) to calculate the second similarity \( Nsim2(u, v) \);
3. Set the threshold \( \beta \) and penalty coefficient \( \lambda \), impose a penalty on the results of the first two steps, and get the user’s final prediction score;
4. Generate recommendations from high to low according to the predicted score.

3. Results and discussion
3.1. Datasets
This experiment used the MovieLens-100K data set, which consists of 100,000 ratings (1-5) from 943 users on 1682 movies and simple demographic info for the users (age, gender, occupation, zip). The attribute information of the movies is represented by 19 types such as Action and Drama. We randomly selected 300 users’ rating data as the data set for our experiment, 80% of which were used as the training set, and the remaining 20% as the test set. The training set and test set were also generated randomly.
3.2. Metrics
This paper used the Mean Absolute Deviation (MAE) to measure the recommendation effect. It is defined as follows:

$$MAE = \frac{\sum_{i=1}^{n}|p_i - r_i|}{n},$$

where $p_i$ represents the target user’s predicted score for item $i$, $r_i$ represents the target user’s true score for item $i$, and $n$ represents the number of items in the test data set. Obviously, the smaller the value of MAE, the more accurate the recommendation effect.

3.3. Choosing parameters $\beta$ and $\lambda$
In our experiment, $\beta$ and $\lambda$ were respectively set to 0, 0.2, 0.4, 0.6, 0.8, 1, the number of neighbor user sets $k$ and the number of items neighbor sets $n$ both took the value 50 when calculating MAE. The experimental results are shown in table 1.

It can be seen from table 1 that when $\beta$ was 0 and 0.2, as $\lambda$ changed, the two changing trends of MAE were exactly the same, and was lower than other $\beta$ values’ MAE. When $\lambda$ was 0.6, the MAE value reached the lowest value. So, in this paper, $\beta$ and $\lambda$ were set to value 0.2 and 0.6 respectively.

| $\lambda$ | $\beta$ | 0   | 0.2 | 0.4 | 0.6 | 0.8 | 1   |
|-----------|---------|-----|-----|-----|-----|-----|-----|
| 0         | 1.041   | 1.041 | 1.042 | 1.042 | 1.042 | 1.042 |
| 0.2       | 0.894   | 0.894 | 0.903 | 0.902 | 0.902 | 0.931 |
| 0.4       | 0.768   | 0.768 | 0.784 | 0.783 | 0.783 | 0.821 |
| 0.6       | 0.711   | 0.711 | 0.736 | 0.735 | 0.735 | 0.767 |
| 0.8       | 0.737   | 0.737 | 0.77  | 0.769 | 0.769 | 0.782 |
| 1         | 0.787   | 0.787 | 0.828 | 0.827 | 0.827 | 0.822 |

3.4. Results
In order to verify the effectiveness of the algorithm we proposed, two other recommendation algorithms were selected for comparison: the BC’ algorithm [5], Collaborative Filtering algorithm combining filling and improving similarity (CICF) [10]. In the calculation of the interpolation phase, the number of users’ nearest neighbors $k$ was set to 50 [13], in the predicting the user’s rating of items phase, the number of items’ nearest neighbors $n$ was set to 20, 30, 40, 50, 60, and the number of users neighbor sets $k$ was set to 20, 30, 40, 50, 60. The final experimental results are shown in figure 1.
It can be seen from figure 1 that the larger the value of $k$, the higher the prediction accuracy of all the three algorithms. The larger the value of $n$, the higher the prediction accuracy of the algorithm we proposed. Furthermore, when $k$ was the same, our algorithm had the lowest MAE value among the three algorithms. When $k = 60$ and $n = 60$, the MAE of the proposed algorithm was 12% higher than the BC’ algorithm and 3.7% higher than the CICF algorithm.

4. Conclusion

Aiming at the problem that item information is not fully utilized, the collaborative filtering algorithm based on joint interpolation is proposed in this paper, takes both user information and item information into consideration during the user rating interpolation and predicting stages. Empirical results show that the proposed algorithm can achieve high accuracy. But took a closer look at the experiment results, we found that the score predicted by the proposed algorithm tends to be more intermediate, such as 3 points in the MovieLens data set scoring standard; when the user’s rating of the item tends to the left and right extremes, such as 1 point or 5 points, there will be a big deviation between the predicted final score and the actual score. This is where the proposed algorithm needs to be improved in the future.

References

[1] George, G., Haas, M.R., Pentland, A. (2014) Big Data and Management. Academy of Management Journal, 27, 321-326.
[2] Zhang, H.B., Xue, F., Liu, K. (2021) Collaborative Filtering Recommendation Algorithm Based on Semi-Autoencoder. Computer Engineering, 47(03):125-130.
[3] Gantz, J., Reinsel, D. (2012) The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East.
[4] Jiang, L., Cheng, Y., Li, Y., Jing, L., Yan, H., Wang, X. (2018) A trust-based collaborative filtering algorithm for e-commerce recommendation system. Journal of Ambient Intelligence and Humanized Computing, 1-12.
[5] Wang, W., Zhou, G. (2020) Collaborative filtering recommendation algorithm based on improved Bhattacharyya coefficient. Application Research of Computers, v.37; No.350(12), 55-57.
[6] Charu C.A. (2018) Recommender Systems: The Textbook. China Machine Press, Beijing.
[7] Huang, X., Zhao, W. (2019) Recommendation System and Deep Learning. Tsinghua University Press, Beijing.

[8] Bokde, D.K., Girase, S., Mukhopadhyay, D. (2015) An item-based collaborative filtering using dimensionality reduction techniques on mahout framework. Computer Science, 134(3), 561-576.

[9] Deng, A.L., Zhu, Y.Y., Shi, B.L. (2003) A collaborative filtering recommendation algorithm based on item rating prediction. Journal of Software, 14(9):1621-1628.

[10] Xing, C.z., Jin Y. (2019) Collaborative Filtering algorithm combining filling and improving similarity. Application Research of Computers, 36(6):1643-1645.

[11] Zou, Y., WU, H.Z., Jiang, Y.Z., Zhao, Y.D. (2020) Recommendation algorithm based on filling method and multi-weight similarity. Application Research of Computers, 37(12):3578-3581+3598.

[12] Francesco, R., Lior, R., Bracha, S. (2018) Recommender Systems Handbook. China Machine Press, Beijing.

[13] Koren, R. (2012) Seventh IEEE International Conference on Data Mining Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights.