Research Article

Implementation of English “Online and Offline” Hybrid Teaching Recommendation Platform Based on Reinforcement Learning

Danling Dong¹ and Libo Wu²

¹Zhejiang Gongshang University, Hangzhou College of Commerce, Hangzhou 310000, China
²Hebei Vocational University of Technology and Engineering, Xingtai 054000, China

Correspondence should be addressed to Danling Dong; 1180080@zjhzcc.edu.cn

Received 12 August 2021; Revised 5 September 2021; Accepted 17 September 2021; Published 30 September 2021

Academic Editor: Jian Su

Copyright © 2021 Danling Dong and Libo Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

At present, there is a serious disconnect between online teaching and offline teaching in English MOOC large-scale hybrid teaching recommendation platform, which is mainly due to the problems of cold start and matrix sparsity in the recommendation algorithm, and it is difficult to fully tap the user’s interest characteristics because it only considers the user’s rating but neglects the user’s personalized evaluation. In order to solve the above problems, this paper proposes to use reinforcement learning thought and user evaluation factors to realize the online and offline hybrid English teaching recommendation platform. First, the idea of value function estimation in reinforcement learning is introduced, and the difference between user state value functions is used to replace the previous similarity calculation method, thus alleviating the matrix sparsity problem. MX_he learning rate is used to control the convergence speed of the weight vector in the user state value function to alleviate the cold start problem. Second, by adding the learning of the user evaluation vector to the value function estimation of the state value function, the state value function of the user can be estimated approximately and the discrimination degree of the target user can be reflected. Experimental results show that the proposed recommendation algorithm can effectively alleviate the cold start and matrix sparsity problems existing in the current collaborative filtering recommendation algorithm and can dig deep into the characteristics of users’ interests and further improve the accuracy of scoring prediction.

1. Introduction

MOOC is the abbreviation of large-scale online courses, which has attracted widespread attention from academic circles for its advantages of large-scale and openness. Large-scale means that there is no limit to the number of people in the course, and a course can be offered to countless people and can be watched repeatedly. Open means that the course has no access conditions and entry threshold, and anyone can use the learning resources free of charge anywhere [1–4]. As an innovation of traditional classroom, MOOC curriculum breaks the monopoly of high-quality learning resources and provides conditions and possibilities for realizing educational equity.

The mixed teaching of online teaching and offline teaching under MOOC mode not only plays an important role in stimulating students’ learning interest, improving their learning enthusiasm, cultivating students’ learning consciousness, and ability of unity and cooperation but also can effectively solve the problem of time allocation between theoretical explanation and practical practice and at the same time strengthen the connection between teachers and students and improve the teaching efficiency. Online teaching includes online video courses, online homework, and online communication and discussion [4–6]. Classroom teaching is composed of three parts: group report, classroom answering, practice, and classroom assessment. At present, this teaching mode has been applied in ideological and political courses and English courses, and the teaching effect has been well received. MOOC mode unifies students’ online and offline and brings them into the credit range, which is more stimulating for students’ learning.
However, there is a serious disconnect between online teaching and offline teaching in English MOOC large-scale mixed teaching recommendation platform, which is because most of the mainstream English MOOC teaching recommendation platforms use collaborative filtering recommendation algorithm to realize personalized recommendation [7–9]. At present, collaborative filtering recommendation algorithm is the best and most widely used algorithm. The algorithm calculates the similarity between target users or objects and predicts the object preference of target users according to the evaluation of objects by users with similar interests. Collaborative filtering recommendation algorithm can be divided into neighborhood-based collaborative filtering recommendation and model-based collaborative filtering recommendation [10–12]. Neighborhood-based collaborative filtering calculates the similarity between target users and user groups to get users with similar interests and hobbies and then predicts the preference of target users for candidate items according to the evaluation status of these users. However, on the one hand, the collaborative filtering algorithm has problems such as cold start and matrix sparsity. On the other hand, the current recommendation algorithms often only consider the user’s rating of items but ignore the user’s personalized evaluation of items, so it is difficult to fully tap the user’s interest characteristics.

Based on the research of recommendation algorithms at home and abroad, aiming at the problems existing in current recommendation algorithms, this study improves the traditional collaborative filtering recommendation algorithm in two aspects. The main contributions of this paper are as follows:

1. To introduce the idea of reinforcement learning, this paper proposes to measure the similarity between users by comparing the state value functions between users instead of the previous similarity calculation method to alleviate the matrix sparsity problem and solve the cold start problem by controlling the convergence speed of the weights in the state value functions.

2. User evaluation factors are added on the basis of reinforcement learning recommendation algorithm. By adding the learning of user evaluation vector to the value function estimation of state value function, the state value function of users can be estimated approximately, and the discrimination of target users can be reflected. Adding the learning of the user’s internal evaluation vector to the weight update can improve the convergence effect of the user’s state value function.

3. Finally, according to the previously mentioned research content, we design and implement the reinforcement learning recommendation algorithm based on user evaluation and compare it with several popular collaborative filtering recommendation algorithms in the prediction error of the recommended content to prove its personalized recommendation effect.

The rest of the paper is organized as follows. In Section 2, a literature review is studied in detail, while Section 3 provides the detailed methodology. Section 4 provides detailed results and discussion. Finally, the paper is concluded in Section 5.

2. Literature Review

At present, many studies have tried to improve collaborative filtering algorithm. For example, Najafabadi et al. [13] proposed an algorithm model of mixed recommendation by adjusting the weights of collaborative filtering algorithm and content-based algorithm, so that the recommendation system can run different recommendation algorithms by adjusting the weights under different conditions. Wu et al. [14] obtained the user's comments on articles by learning the text data and proposed a hybrid recommendation system based on knowledge. Yang et al. [15] put forward an algorithm of connecting two recommendation modes in series. First, collaborative filtering method is used to get the recommended items of target users, and then, the content-based recommendation algorithm is used to model the target users, and the items in the recommended items set that are inconsistent with the user model are eliminated, thus improving the recommendation effect.

Reinforcement learning, together with supervised learning and unsupervised learning, is an important branch of machine learning method. Supervised learning expects to predict unlabeled data outside the training set by learning labeled data in the training set, while unsupervised learning expects to discover hidden information in the data and make predictions through the hidden information. Reinforcement learning is also learning data that is not marked in the training set, but it will get a delayed reward signal. The agent of reinforcement learning can adjust itself according to the reward signal in order to expect more rewards.

Polvara et al. [16] research results show that the idea of reinforcement learning is to maximize the sum of long-term rewards as the goal, through the independent exploration of the unknown environment, in a certain state and action under the influence of the environment and get the corresponding reward and then get more reward adjustment strategy, ultimately in the process of continuous interaction with the environment learning to the optimal strategy.

3. Methodology

3.1. Calculated Value Function Model Based on Reinforcement Learning

The running process of the reinforcement learning algorithm is a state-action-reward-new-state transfer sequence process, that is, the agent according to the action taken from a state to obtain environmental feedback reward and transfer to a new state, and the process is repeated until the end of the state. Since the state of the agent at the next moment only depends on the state of the agent at the current moment and the action to be taken, the process has Markov property. The reinforcement learning median function estimation method is an extremely important
method; that is, the optimal strategy is obtained through the state value function or the action value function of the agent.

In the application environment of recommendation scenarios, what we need to compare in this paper is the similarity of value functions between users, and we pay more attention to the user’s own state. Therefore, we use the user’s state value function at each moment to approximately estimate the user’s current state. A value function estimation method evaluates a value function of a state by using a sum of linear product of a set of features and corresponding weights [17]:

\[ V_w(s) = \sum_{i=1}^{M} w_i \phi_i(s), \tag{1} \]

where \( \phi_i(s) \) represents the feature function of state \( s \), \( M \) represents the number of feature functions, and \( w \) represents the weight vector corresponding to each feature.

The gradient descent method is used to find the optimal solution \( w^* \), so that the state value function converges to the local optimum. The gradient descent method is one of the most popular methods for solving practical problems.

The state value function of the agent at each time \( t \) is approximately estimated using the value function estimation idea in the reinforcement learning method, and the weight of each state is adjusted in real time by using the gradient descent method so that the estimation of the value function is more accurate. The reward function \( R \) obtained for each step of the agent is taken as a characteristic function of the state. In the actual application scenario of the recommendation algorithm, the user’s rating of a certain item at a certain moment can be regarded as a state of the user, and the user’s rating vector of the item can be taken as a reward function \( R \), that is, a characteristic function of the user in the state [18]. Thus, the user’s value function is estimated as follows:

\[ V_t(s) = \sum_{i=1}^{n} w_i (i) R_i (s), \tag{2} \]

where \( V_t(s) \) represents the state value function vector of the user in the state \( s \) at time \( t \), \( R_i \) represents a scoring vector of the user on an article, and \( w_i \) represents a weight corresponding to each feature vector at time \( t \).

According to the gradient descent method, the updated formula of \( w_i \) can be converted into

\[ w_i = w_i + \alpha (r_{i+1} + \gamma V(s_{i+1}) - V(s_i)) \nabla_w V(s_i), \tag{3} \]

In this formula, \( \alpha \) represents the learning rate (generally between 0 and 1), \( r_{i+1} \) represents the immediate reward value obtained by the user at time \( i+1 \) (in the recommended scenario, it represents the user’s scoring value of the article at time \( i+1 \)), \( \gamma \) represents the time discount factor, \( V(s_i) \) represents the state value function of the user’s state \( s \) at time \( i \), and \( \nabla_w V(s_i) \) represents the gradient of the user’s state value function to the weight \( w \).

In reinforcement learning recommendation algorithm, it is through the adjusted value that the cold start problem of collaborative filtering algorithm is alleviated. The key steps of approximate estimation and updating state value function are shown in Figure 1.

In order to alleviate the cold start and matrix sparsity problems of collaborative filtering algorithm, the idea of reinforcement learning median function estimation is adopted to replace the similarity calculation method of collaborative filtering algorithm by calculating the user’s state value function, and the similarity between users is compared according to the user’s state value function to make neighbor recommendation. The similarity of state value function vector can be calculated with all users by calculating the similarity of state value function, thus effectively alleviating the problem that it is difficult to calculate the similarity between the target user and other users when the matrix sparsity is high.

Because the state value function is a vector value, which is difficult to compare directly, the user with the highest similarity with the target user’s state value function is found by cosine similarity calculation. The calculation formula is as follows:

\[ \text{sim}_{ij} = \frac{\nabla V(s_i) \cdot \nabla V(s_j)}{|\nabla V(s_i)| |\nabla V(s_j)|}, \tag{4} \]

where \( \nabla V(s_i) \) and \( \nabla V(s_j) \) represent the final state value functions of user \( i \) and user \( j \), respectively.

### 3.2. Evaluation Criteria

Root mean square error (RMSE) and mean absolute percentage error (MAPE) are used to measure the error between the predicted score and the true score of recommended courses [19]:

\[ \text{RMSE} = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left[ \text{prediction}(i) - \text{real}(i) \right]^2}, \tag{5} \]

\[ \text{MAPE} = \frac{100\%}{M} \sum_{i=1}^{M} \left| \frac{\text{real}(i) - \text{prediction}(i)}{\text{real}(i)} \right|, \]

where \( m \) represents the total number of recommended courses for users, \( \text{prediction}(i) \) represents the prediction score of recommended courses \( i \), and \( \text{real}(i) \) represents the actual score of courses \( i \) by users.

### 3.3. Application of User Evaluation in Reinforcement Learning Method

The current recommendation algorithm’s strategy on user evaluation application is to build user-user evaluation matrix and user evaluation and item matrix. The user-item matrix is obtained by multiplying the two, and the target user is selected to recommend the item with the highest item score. The principle of traditional user evaluation recommendation algorithm is shown in Figure 2.

Firstly, the user-user evaluation matrix is constructed:
where $u$ represents user (1, 2, ..., $m$) and $e$ represents user evaluation (1, 2, ..., $s$). The value of $L_{ue}$ in the matrix indicate whether the user $m$ has used the user evaluation $s$ (if yes, the value is 1; if no, it is 0). Second, a user evaluation-item matrix is constructed:

$$L_{mx} = \begin{pmatrix}
L_{u_{(1)}, e_{(1)}} & \cdots & L_{u_{(1)}, e_{(s)}} \\
\vdots & \ddots & \vdots \\
L_{u_{(m)}, e_{(1)}} & \cdots & L_{u_{(m)}, e_{(s)}}
\end{pmatrix},$$

(6)

where $i$ represents item (1, 2, ..., $n$). The value of $L_{ie}$ in the matrix indicate whether the item $n$ has been used for user evaluation $s$ (if yes, the value is 1; if no, it is 0). The user-item matrix can be obtained by multiplying two matrices:

$$L_{mxn} = \begin{pmatrix}
L_{u_{(1)}, i_{(1)}} & \cdots & L_{u_{(1)}, i_{(n)}} \\
\vdots & \ddots & \vdots \\
L_{u_{(m)}, i_{(1)}} & \cdots & L_{u_{(m)}, i_{(n)}}
\end{pmatrix}.$$

(8)

According to the user-item matrix, the item with the highest score corresponding to the target user in the matrix can be selected and recommended to the corresponding target user.

However, the algorithm also failed to fully mine the information of user evaluation itself and simply marked 0 and 1 according to the user evaluation. It failed to make full use of the user’s likes and dislikes of articles from the user evaluation and failed to divide the user’s interests well, which made it difficult to reflect the distinction between the target user and the rest of the users.

Therefore, this study proposes to take the user evaluation vector as a part of the user state value function in reinforcement learning to reflect the user interest, which is
Each dimension in the user evaluation vector represents the user's use of the user evaluation, which is not simply represented by 0 and 1 as the matrix multiplication previously mentioned, but represented by the weight of the evaluation used by the user among all users:

\[ \varphi_u(e_i) = \frac{1 + e_{ui}}{\sum_{u=1}^{m} 1 + e_{ui}} \]  

(9)

where \( e_{ui} \) represents the number of times that user \( u \) uses the \( i \)-th dimension of user evaluation vector, and \( m \) represents the total number of users. According to formula (8), the frequency of each user evaluation \( e \) used by user \( u \) in the whole user group can be obtained, which can reflect the discrimination degree of a single user's user evaluation in the whole user group.

\( \varphi_u \) is regarded as a part of the user state value function to reflect the discrimination degree of user \( u \) in the user group. On the other hand, in Section 3.1, the weight vector \( w \) is updated by gradient descent method, and the learning of user evaluation is also needed in the method of making the user state value function reach the optimum. Different from the use of user evaluation in (9), in order to learn the weight of \( w \) better, it is necessary to learn the weight of each user evaluation within the user.

\[ w_{(t_i,j)} = w_{t_{i,j}} + \alpha \frac{N_i}{\sum_{j=1}^{N_j} (r_{t_i} + \gamma V_{t_i}(s) - V_{t_i}(s))} \]  

(10)

where \( w_{(t_i,j)} \) represents the \( i \)-th dimension user evaluation weight in the user evaluation vector at \( t_i \), \( r_{t_i} \) represents the user’s scoring value for items at \( t_i \), \( N_i \) represents the \( i \)-th dimension user evaluation times in the user evaluation vector, \( \sum_{j=1}^{n} N_j \) represents the sum of user evaluation times of all dimensions in the user evaluation vector, and \( n \) represents the dimension degree of the user evaluation vector.

It can be seen that the weight of each user evaluation within the user is added on the basis of formula (3) of gradient descent method in Section 3.1, which can reflect the discrimination between each user evaluation within the user. This way of updating weight \( w \) can make the solution process of user state value function pay more attention to the study of user’s internal state. Specifically, by using the weight of a single user’s evaluation times within the user, it can be distinguished from other users’ evaluations within the user.

In order to further improve the performance of recommendation algorithm and personalized recommendation ability, the user evaluation vector is added to the user’s state value function to approximate the user’s current state more accurately:

\[ V_{t_i}(s) = \sum_{i=1}^{n} w_{t_{i}}(i)R_{i}(i)\varphi_{i}(e_i), \]  

(11)

where \( V_{t_i}(s) \) represents the state value function vector of the user at \( t_i \) time \( s \), \( R_{i}(i) \) represents the \( i \)-th dimension of the scoring vector, \( \varphi_{i}(e_i) \) represents the \( i \)-th dimension of the user evaluation vector as shown in formula (9), and \( w_{t_{i}} \) represents the weight vector corresponding to the feature vector composed of the scoring vector and the user evaluation vector.

By adding user evaluation vector to describe the frequency of user evaluation in the whole user group, so as to reflect the discrimination between users and other users, the user’s state value function can be estimated approximately more accurately. Adding the learning of users’ internal weights can adjust the gradient and make the state value function converge to the optimal value more easily. Finally, the user closest to the target user’s state value function is calculated by cosine similarity formula (4), and the favorite items are selected for neighbor recommendation. The specific recommendation process is shown in Figure 3.

### 3.4. Implementation of Hybrid Teaching Recommendation Platform

In MOOC mode, the evaluation of students’ learning effect includes the following parts: 20% video learning progress and online quiz, 30% classroom performance (classroom language skills training performance and group report performance), 10% homework submission and quality, and 40% final exam. In addition, in the specific teaching practice, teachers can adjust the proportion of tests according to different class types, such as oral English class, writing class, reading class, and listening class, guide the emphasis of teaching while combining with teaching practice, change the single test method in traditional teaching, give students more opportunities to show their comprehensive ability, explore their potential, and realize new evaluation in online education environment. The operation mode of the hybrid teaching recommendation platform is shown in Figure 4.

### 4. Experimental Results and Analysis

#### 4.1. Experimental Data and Pretreatment

The experiment is carried out on the MOOC learning platform of Chinese universities, and the simulation tool is MATLAB7.8. The English curriculum data files of 10 schools are randomly selected on the learning platform and preprocessed. After data preprocessing, it includes five fields: user id, course id, user rating (0 to 5 points, increasing according to half points), user evaluation (filtered from tags), and time when users submit rating evaluation. The characteristics of 10 datasets after pretreatment in this experiment are shown in Table 1.

| Dataset | Number of Users | Number of Courses | Average Rating |
|---------|-----------------|------------------|---------------|
| Dataset A | 1000 | 500 | 3.5 |
| Dataset B | 1500 | 750 | 4.0 |
| Dataset C | 2000 | 1000 | 4.5 |

It can be seen from Table 1 that the number of user evaluations is slightly more than the number of user ratings, because users often only score courses once, but the user evaluations made may contain many words that reflect the likes and dislikes of users, and these words can be used as one dimension of the user evaluation vector.

#### 4.2. Comparison and Analysis of Experimental Results

Root mean square error (RMSE) and mean absolute percentage error (MAPE) are used to jointly evaluate the deviation between the predicted score and the true score of
recommended courses. First, according to RMSE, the specific values of time discount factor $c$ and learning rate $\alpha$ in each dataset are determined. After determining the optimal values, the performance of the proposed algorithm is compared with several popular collaborative filtering recommendation algorithms to fully verify its effectiveness in improving the accuracy of personalized recommendation of users.

Because time discount factor $c$ and learning rate $\alpha$ directly affect the convergence effect of weight $w$, these two parameters need to be slightly adjusted on different scale datasets to make $w$ achieve better convergence effect. In order to make the estimated value of the next state value function better approximate to the true value of the current state value function, the value of $c$ is between 0.5 and 1.0, and the value of $\alpha$ is between 0.1 and 1.0. The best combination value of the two is determined by the recommended performance.

For dataset A, the influence of combination value of $\alpha$ and $c$ on the recommended algorithm is shown in Table 2. For dataset B, the influence of combination value of $\alpha$ and $c$ on the recommended algorithm is shown in Table 3.
Taking datasets A and B as examples, the influence surfaces of $\alpha$ and $c$ combination values on the recommendation algorithm are shown in Figure 5.

It can be seen from Figure 5 that, for dataset A, when $\alpha = 0.8$ and $c = 0.8$, the accuracy of the reinforcement learning recommendation algorithm based on user evaluation reaches the best. For dataset B, the accuracy is optimal when $\alpha = 0.7$ and $c = 0.7$. According to the results, when the dataset is small, $\alpha$ should be increased to converge to the optimum faster. When the dataset is large, $\alpha$ should be appropriately reduced to better control the law of convergence effect.

After determining the optimal combination of $\alpha$ and $c$ on datasets A and B, the proposed algorithm is compared with user-based collaborative filtering algorithm [20], item-based collaborative filtering algorithm [21], user-based K-nearest neighbor recommendation algorithm [22], and trust-based matrix decomposition recommendation algorithm [23, 24], respectively. The comparison results of RMSE are shown in Figure 6.

It can be seen from Figure 6 that the RMSE of the proposed recommendation algorithm between the predicted score and the true score of the recommended course on datasets A and B is the smallest, indicating that the absolute

| Dataset name | Number of users | Number of courses | Average number of user ratings | Average number of user evaluations | Number of data | Sparsity (%) |
|--------------|----------------|------------------|-------------------------------|-----------------------------------|----------------|--------------|
| A            | 316            | 6                | 23.71                         | 24.68                             | 52510          | 97.74        |
| B            | 467            | 4                | 20.21                         | 22.38                             | 85260          | 98.70        |
| C            | 354            | 3                | 21.06                         | 25.32                             | 45060          | 96.36        |
| D            | 398            | 6                | 22.84                         | 24.36                             | 55710          | 97.06        |
| E            | 539            | 4                | 26.14                         | 24.02                             | 61280          | 96.02        |
| F            | 610            | 6                | 24.02                         | 25.28                             | 115060         | 97.82        |
| G            | 496            | 7                | 23.68                         | 23.35                             | 45370          | 97.11        |
| H            | 503            | 8                | 22.10                         | 22.06                             | 50310          | 97.92        |
| I            | 469            | 5                | 25.71                         | 25.34                             | 49580          | 96.35        |
| J            | 385            | 4                | 23.16                         | 24.61                             | 91130          | 97.24        |

| $\alpha$ | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|----------|-----|-----|-----|-----|-----|-----|
| 0.1      | 0.9173 | 0.9103 | 0.9072 | 0.9049 | 0.9051 | 0.9089 |
| 0.2      | 0.9148 | 0.9087 | 0.9027 | 0.9003 | 0.9013 | 0.9043 |
| 0.3      | 0.9116 | 0.9044 | 0.8983 | 0.8971 | 0.8977 | 0.9018 |
| 0.4      | 0.9108 | 0.9019 | 0.8961 | 0.8942 | 0.895 | 0.9004 |
| 0.5      | 0.9084 | 0.8983 | 0.8934 | 0.8915 | 0.8929 | 0.8971 |
| 0.6      | 0.9045 | 0.8952 | 0.8903 | 0.8873 | 0.8897 | 0.8933 |
| 0.7      | 0.9027 | 0.8914 | 0.8871 | 0.8857 | 0.8863 | 0.8916 |
| 0.8      | 0.9012 | 0.8896 | 0.8847 | 0.8838 | 0.8851 | 0.8873 |
| 0.9      | 0.9041 | 0.8903 | 0.8864 | 0.8847 | 0.887 | 0.891 |

| $\gamma$ | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|----------|-----|-----|-----|-----|-----|-----|
| 0.1      | 0.8934 | 0.892 | 0.8912 | 0.8913 | 0.8923 | 0.8973 |
| 0.2      | 0.8917 | 0.8903 | 0.8884 | 0.8906 | 0.891 | 0.8924 |
| 0.3      | 0.8894 | 0.8874 | 0.8858 | 0.888 | 0.8893 | 0.8901 |
| 0.4      | 0.8872 | 0.8859 | 0.8821 | 0.8847 | 0.8864 | 0.8879 |
| 0.5      | 0.8869 | 0.8815 | 0.8803 | 0.8826 | 0.8842 | 0.8854 |
| 0.6      | 0.885 | 0.8793 | 0.8777 | 0.8799 | 0.8831 | 0.8834 |
| 0.7      | 0.8831 | 0.8772 | 0.8754 | 0.8782 | 0.8793 | 0.8827 |
| 0.8      | 0.8842 | 0.8786 | 0.876 | 0.8793 | 0.8814 | 0.8839 |
| 0.9      | 0.8878 | 0.8823 | 0.8809 | 0.8816 | 0.884 | 0.885 |
error between the predicted score and the true value of the recommended course is the smallest in the reinforcement learning recommendation algorithm based on user evaluation.

Then, we compare MAPE [25–28] with the previously mentioned four popular collaborative filtering algorithms on two datasets A and B, so as to investigate the accuracy of each algorithm in prediction score more comprehensively. The experimental results are shown in Figure 7.

It can be seen from Figure 7 that the MAPE of the reinforcement learning recommendation algorithm model based on user evaluation is the smallest between the predicted score and the true score of the recommended course through the divided data of different scales on the two datasets A and B. Compared with other algorithms, the proposed algorithm not only performs well in each sparse matrix but also has great advantages (no matter absolute error or relative error) in datasets with small data scale.
5. Conclusion

In this paper, the idea of value function estimation in reinforcement learning is introduced, and the similarity between users is evaluated through the state value function between users (which is different from the traditional similarity calculation method), thus effectively alleviating the matrix sparsity problem and alleviating the cold start problem by controlling the convergence speed of the control value. At the same time, the user evaluation vector is introduced, and the user’s interests and hobbies are mined by the proportion of user evaluation in the user group and within the user, so as to improve the personalized recommendation effect. Experiments on the English MOOC hybrid teaching recommendation platform show that the error of the reinforcement learning recommendation algorithm based on user evaluation in predicting the score of courses is smaller than other existing collaborative filtering recommendation algorithms in all datasets, thus achieving better recommendation performance.

Data Availability

The dataset used in this paper are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partially supported by the “Blended Learning Reseach under the Background of Internet,” China (grant no. EJYB 2017–061).

References

[1] M. J. Puzziferro and K. Shelton, “A model for developing high-quality online courses: integrating a systems approach with learning theory,” Online Learning, vol. 12, pp. 3–4, 2019.
[2] V. Swigart and Z. Liang, “Digital resources for nursing education: open courseware and massive open online courses,” International Journal of Nursing Science, vol. 3, no. 3, pp. 307–313, 2016.
[3] T. A. Baran, R. G. Baraniuk, A. V. Oppenheim, P. Prandoni, and M. Vetterli, “MOOC Adventures in Signal Processing: bringing DSP to the era of massive open online courses,” IEEE Signal Processing Magazine, vol. 33, no. 4, pp. 62–83, 2016.
[4] J. R. MacKay, F. M. Langford, and N. Waran, “Massive open online courses as a tool for global animal welfare education,” Journal of Veterinary Medical Education, vol. 43, no. 3, pp. 11–18, 2016.
[5] B. Chen, A. deNoyelles, J. Zydney, and K. Patton, “Creating a community of inquiry in large-enrollment online courses: an exploratory study on the effect of protocols within online discussions,” Online Learning, vol. 21, pp. 2590–2594, 2017.
[6] M. Then, B. Wallenborn, B. R. Ianniello, D. Binh Vu, M. Fuchs, and M. L. Hemmje, “Innovative authoring tools for online-courses with assignments - integrating heterogeneous tools of e-learning platforms into hybrid application solutions,” International Journal of Emerging Technologies in Learning, vol. 11, no. 12, pp. 2–20, 2016.
[7] J. Kasch, P. V. Rosmalen, and M. Kalz, “A framework towards educational scalability of open online courses,” Journal of Universal Computer Science, vol. 23, no. 9, pp. 845–867, 2017.
[8] J. C. Dunlap, G. Verma, and H. L. Johnson, “Presence+Experience: a framework for the purposeful design of presence in online courses,” TechTrends, vol. 60, no. 2, pp. 145–151, 2016.
[9] J. Paterson, K. Hughes, L. Steer et al., “Massive open online courses (MOOCs) as a window into the veterinary profession,” The Veterinary Record, vol. 180, no. 7, p. 179, 2017.
[10] P. J. Fidaldo, “Reaching students in online courses using alternative formats,” International Review of Research in Open and Distance Learning, vol. 18, no. 2, pp. 89–94, 2017.

[11] P. Corke, E. Greener, and R. Philip, “An innovative educational change: massive open online courses in robotics and robotic vision,” IEEE Robotics and Automation Magazine, vol. 23, no. 2, pp. 81–89, 2016.

[12] O. Pilli and W. Admiraal, “Students’ learning outcomes in massive open online courses (MOOCs): some suggestions for course design,” Yuksekogretim Dergisi, vol. 7, no. 1, pp. 46–71, 2017.

[13] M. K. Najafabadi, M. N. r. Mahrin, S. Chuprat, and H. M. Sarkan, “Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data,” Computers in Human Behavior, FEB, vol. 67, pp. 113–128, 2017.

[14] X. Wu, B. Cheng, and J. L. Chen, “Collaborative filtering service recommendation based on a novel similarity computation method,” IEEE Transactions on Services Computing, vol. 10, no. 99, pp. 21–28, 2017.

[15] X. Yang, C. Liang, M. Zhao et al., “Collaborative filtering-based recommendation of online social voting,” IEEE Transactions on Computational Social Systems, vol. 4, no. 1, pp. 1–13, 2017.

[16] R. Polvara, S. Sharma, J. Wan, A. Manning, and R. Sutton, “Autonomous vehicular landings on the deck of an unmanned surface vehicle using deep reinforcement learning,” Roboticica, vol. 37, no. 11, pp. 1867–1882, 2019.

[17] M. Nilashi, O. Ibrahim, and K. Bagherifard, “A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques,” Expert Systems with Applications, vol. 92, pp. 507–520, 2017.

[18] Z. Xian, Q. Li, G. Li, and L. Li, “New collaborative filtering algorithms based on SVD++ and differential privacy,” Mathematical Problems in Engineering, vol. 2017, Article ID 1975719, 14 pages, 2017.

[19] Y. Hu, W. Shi, H. Li, and X. Hu, “Mitigating data sparsity using similarity reinforcement-enhanced collaborative filtering,” ACM Transactions on Internet Technology, vol. 17, no. 3, pp. 1–20, 2017.

[20] Z. Li, H. Li, and Y. Feng, “Research on big data mining based on improved parallel collaborative filtering algorithm,” Cluster Computing, vol. 22, pp. 1–10, 2019.

[21] D. Li, C. Chen, Q. Lv et al., “An algorithm for efficient privacy-preserving item-based collaborative filtering,” Future Generation Computer Systems, vol. 55, pp. 311–320, 2016.

[22] X. Luo, Y. Xia, Q. Zhu, and Y. Li, “Boosting the K-Nearest-Neighborhood based incremental collaborative filtering,” Knowledge-Based Systems, vol. 53, pp. 90–99, 2013.

[23] W. Yuan, K. Velswamy, and B. Huang, “A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems,” Processes, vol. 5, pp. 46–51, 2017.

[24] A. Mkrichian, J. Aylward, P. Dayan, J. P. Rosier, and O. J. Robinson, “Modeling avoidance in mood and anxiety disorders using reinforcement learning,” Biological Psychiatry, vol. 82, no. 7, pp. 532–539, 2017.

[25] A. Sallab, M. Abdou, E. Perot, and S. Yogamani, “Deep reinforcement learning framework for autonomous driving,” Electronic Imaging, vol. 2017, no. 19, pp. 70–76, 2017.

[26] H. Ying, Z. Zhang, F. R. Yu et al., “Deep reinforcement learning-based optimization for cache-enabled opportunistic interference alignment wireless networks,” IEEE Transactions on Vehicular Technology, vol. 66, Article ID 10433, 2017.