

Research Article

Ideological and Political Education Recommendation System Based on AHP and Improved Collaborative Filtering Algorithm

Nan Wang 

School of Xingjian Academy, Xijing University, Xi'an, Shaanxi 710123, China

Correspondence should be addressed to Nan Wang; 13991115103@163.com

Received 26 October 2021; Revised 11 November 2021; Accepted 18 November 2021; Published 14 December 2021

Academic Editor: Le Sun

Copyright © 2021 Nan Wang. 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.

Aiming to solve the problem that ideological and political education courses in universities are not targeted enough and cannot form personalized recommendations, this paper proposes an ideological and political education recommendation system based on analytic hierarchy process (AHP) and improved collaborative filtering algorithm. Firstly, considering the time effect of student scoring, the recommendation model is transformed into Markov decision process. Then, by combining the collaborative filtering algorithm with reinforcing learning rewards and punishments, an optimization model of student scoring based on timestamp information is constructed. To quantify the degree of students' preference for courses, the analytic hierarchy process is used to convert the students' behavior data into the preference value of courses. To solve the problem of data scarcity, the missing values are predicted by the prediction score rounding filling and the optimization boundary completion method. Experimental results show that the feasibility of the proposed system is verified, and the system has vital accuracy and convergence performance. The ideological and political education recommendation system proposed in this paper has important reference significance for promoting ideological and political education in the era of big data.

1. Introduction

As the main channel to carry out ideological and political work in colleges and universities, ideological and political education curriculum is a critical way to practice the mechanism of education in colleges and universities. Ideological and political education courses run through the whole process of higher education teaching and are an essential system for universities to cultivate high-quality talents [1]. At present, ideological and political education in colleges and universities mostly adopts the way of large classes and open classes for collective teaching. There are many problems in this way, such as single form, weak pertinence, lack of synergistic effect, and being unable to form personalized collaborative education mechanism. Therefore, combining with the characteristics of students, it is of great significance to carry out personalized recommendations of ideological and political education courses.

With the continuous development of network and information technology, the amount of network information

data of ideological education courses is also increasing exponentially. Faced with a variety of ideological education courses, how to carry out personalized recommendations needs to be systematically studied. Personalized recommendation can effectively filter unwanted information by analysing users' behavioural preferences through various recommendation algorithms [2]. Recommendation system can actively provide personalized information for users. At present, personalized recommendation has been widely used in social networking [3], news, music, books, and movies [4], such as cloud music [5] and online shopping product recommendation [6].

Recommendation algorithms are mainly divided into the following three categories [7]. In the first category, collaborative filtering recommendation algorithm makes recommendations based on users' intentions [8, 9]. It has achieved significant improvement in recommendation accuracy. However, their interpretation of recommendation results is often not intuitive [10]. In the second category, content-based recommendation algorithm conducts feature

modeling through various available content information [11, 12]. Because the content of an item is easier for the user to understand, it is often intuitive to explain to the user why the item is being recommended. Collecting the required content information under different recommendation backgrounds is a time-consuming task, which becomes the bottleneck of content-based recommendation algorithms. However, the construction of knowledge map can reduce the workload of extracting content information. Therefore, knowledge graph, as an emerging auxiliary data, is attracting the attention of researchers [13]. Deep learning also brings innovative ideas for recommendation system [14]. However, the existing recommendation algorithm based on deep learning only considers the rating data by using matrix decomposition, which inhibits the recommendation effect [15]. In the third category, hybrid recommendation algorithm [16, 17] can often achieve better recommendation effect by combining the advantages of multiple recommendation algorithms.

As a typical recommendation algorithm, collaborative filtering technology mainly includes memory-based and model-based algorithms [18]. The former calculates the similarity by analysing the user-item score matrix and makes the prediction and recommendation based on the similarity. The latter trains a prediction model through the user's network history, operation, and other data. It then uses this prediction model to predict the project score. Many studies have optimized the recommendation effect by improving collaborative filtering algorithms, such as restricted Boltzmann machine [19], K-nearest neighbour algorithm [20], and singular value decomposition (SVD) [21] algorithm. Not only is SVD a mathematical problem, but also it has been successfully applied in many engineering applications. In the recommendation system, it is easy to obtain the full rank decomposition of any matrix by using SVD. Then, it can realize data compression and dimensional reduction. SVD++ [22] further integrates implicit feedback information on the basis of SVD and adopts implicit preference to optimize SVD model with better performance. However, SVD++ and SVD do not consider the impact of time timestamp on recommendation performance, and the actual recommendation effect is related to timestamp to a certain extent. For example, the ratings of users ten years ago on a certain product are different from those of current users, so it is necessary to improve it and optimize the prediction effect.

Considering the time effect of student rating, the recommendation model is transformed into Markov decision process. Then, in order to quantify the degree of students' preference for courses, the analytic hierarchy process is used to convert the students' behaviour data into the preference value of courses. Finally, the accuracy of the recommendation model is improved by constructing a student scoring optimization model integrating timestamp information.

To solve the problem that ideological and political education courses in universities are not targeted enough and cannot form personalized recommendations, this paper proposes an ideological and political education recommendation system. The contribution of our paper is summarized, which is grouped into the following three points.

- (a) The recommendation model is transformed into Markov decision process considering the time effect of student scoring.
- (b) An optimization model of student scoring based on timestamp information is constructed by combining the collaborative filtering algorithm with the process of reinforcing learning rewards and punishments.
- (c) To quantify the degree of students' preference for courses, the analytic hierarchy process is used to convert the students' behavior data into the preference value of courses.

The rest of the paper is organized as follows. In Section 2, modeling for the system is introduced first, and then the process of training and prediction is introduced. Section 3 gives the experimental results and discussion. At last, Section 4 draws a conclusion of this paper.

2. The Proposed Recommendation System

2.1. Modeling for the System. Firstly, the curriculum recommendation system of ideological and political education is modelled, and the high-dimensional sparse matrix is decomposed into low-dimensional matrix by singular value decomposition method. Then, students' behaviour data are converted into course preference values through analytic hierarchy process (AHP). Finally, the recommendation model is transformed into Markov decision process by establishing model mapping.

2.1.1. Singular Value Decomposition. In real life, the student-course matrix is large, but due to the limited interests of students. The grading data of individual students on courses are often small. The core idea of SVD is to decompose a high-dimensional sparse matrix into two low-dimensional matrices. Compared with eigenvalue decomposition, it can only be used for symmetric matrices. SVD can perform full rank decomposition on any $M \times N$ matrix to achieve data compression. However, before SVD is used to decompose the matrix, blank items in the matrix need to be filled to get a dense matrix. Assuming that the matrix before filling is R and after filling is R' , the calculation formula is as follows:

$$R' = URV^T. \quad (1)$$

The calculation formula of SVD algorithm is as follows:

$$\hat{r}_{px} = \mu + d_p + d_x + v_x^T u_p, \quad (2)$$

where \hat{r}_{px} represents the predicted score value, μ represents the average value of the score, d_p and d_x represent the offset amount of student p and course x , respectively, v_x and u_p correspond to the feature vectors of courses and students on each hidden trait, respectively, and T superscript stands for transpose.

If a student scores a course, it means he has seen it. Such behaviour contains certain information, so it can be inferred that the behaviour of grading reflects students' preferences. Accordingly, this preference can be reflected in the model in

the form of implicit parameters to obtain a more accurate model SVD++.

The calculation formula of SVD++ model is as follows:

$$\hat{r}_{px} = \mu + d_p + d_x + v_x^T \left(u_p + \frac{1}{\sqrt{|T(p)|}} \sum_{y \in T(p)} j_y \right), \quad (3)$$

where $T(p)$ is the collection of all courses browsed and evaluated by student p , j_y bias for hidden student-course J 's personal preferences, and the preference degree of student p consists of explicit feedback p_u and implicit feedback $1/\sqrt{|T(p)|} \sum_{y \in T(p)} j_y$.

2.1.2. Preference Calculation Method Based on AHP. To analyze the hidden bias of students' preference for courses, the data of students' browsing course introduction, collecting course, and learning course were firstly collected. Then, the analytic hierarchy process (AHP) is used to calculate the value of students' preference for courses. The questions favoured by students are divided into goal layer, criterion layer, and program layer. Use browsing course profiles, bookmarking courses, and studying courses as criteria for making decisions. The hierarchical model of student interest calculation is shown in Figure 1.

The judgment matrix C is constructed by pairwise comparison of different factors. It is more important to bookmark the course than to browse the course introduction. Learning the course content is more important than collecting the course. Therefore, the scale is set as follows: the ratio of collection course to browsing course is 3; the ratio of learning course to browsing course is 5; the ratio of learning course to collection course is 3. Therefore, the judgment matrix C is

$$C = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{5} \\ 3 & 1 & \frac{1}{3} \\ 5 & 3 & 1 \end{bmatrix}. \quad (4)$$

The maximum eigenvalue of the matrix is

$$\lambda = 3.0414. \quad (5)$$

The eigenvector m of the matrix is

$$m = (0.11, 0.26, 0.63)^T. \quad (6)$$

EX of consistency indicator is

$$EX = \frac{3.0414 - 3}{3 - 1} = 0.0207, \quad (7)$$

and Table 1 shows that the average random consistency index RX is

$$RX = 0.52. \quad (8)$$

The size of the calculated consistency ratio ER is

$$ER = \frac{EX}{RX} = \frac{0.0207}{0.52} = 0.0398 < 0.1. \quad (9)$$

The final result passes the consistency ratio test. Therefore, the weight of browsing course profiles is 0.11. The weight of collecting courses is 0.26, and the weight of learning courses is 0.63.

The degree of students' preference for course operation is quantified. The calculation formula of students' preference intention is as follows:

$$V = 0.11 * U1 + 0.26 * U2 + 0.63 * U3. \quad (10)$$

$U1$ here browses the course introduction for students. $U2$ has a collection of lessons for students. $U3$ studied the course for students. Therefore, the analytic hierarchy process (AHP) is used to convert student behaviour data into course preference values.

2.1.3. Markov Decision-Making Process. Markov decision process is an intuitive and basic construction model in decision theoretic programming, reinforcement learning, and stochastic domain. In this model, the environment is modelled through a set of states and actions that can be used to execute the state of the control system. The goal of controlling the system in this way is to maximize a model's performance criteria. At present, many problems have been successfully modelled by Markov decision process, such as multiagent problem, robot learning control, and game playing problem. Therefore, Markov decision process has become a standard method to solve timing decision problems.

A general Markov decision process is represented by a quintuple, as shown in Figure 2, where S_n represents the state, c_n represents the action, and r_n represents the return function. The agent senses the state information in the current environment and chooses to perform some actions according to the current state. The environment sends a reward or punishment signal to the agent based on the selected action. Based on this reward and punishment signal, the agent moves from one state to the next.

To optimize SVD++ recommendation model by reinforcement learning method, the mapping relationship between recommendation prediction model and Markov decision process should be established first. In order to construct Markov decision process, students' preference scores for courses under different time stamps are converted into quintuples. The following is the mapping relationship between course scoring designed in this paper and Markov decision process.

- (1) For state space S , in this paper, student p scores the course at time n as state $S_n^{(p)}$. Because the score of students on courses in the data set is 5 integers within the range $[1, 5]$, the range of $S_n^{(p)}$ is $[1, 5]$. The state $S_n^{(p)}$ under all timestamps constitutes the state space S .
- (2) For action space C , student U looks at the course under time n and gives a grade $S_n^{(p)}$, which will affect

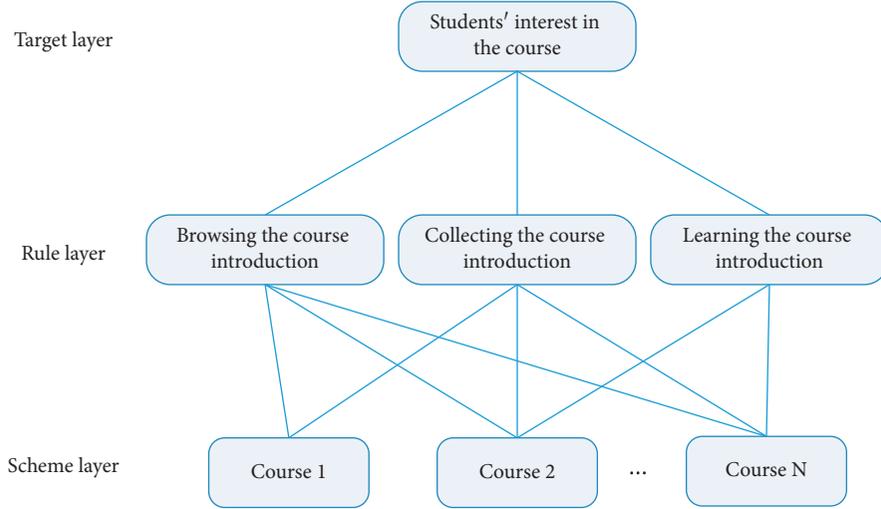


FIGURE 1: The hierarchical model of student interest calculation.

TABLE 1: Reference table of average random uniformity index RX value.

Order number	RX
1	0
2	0
3	0.52
4	0.89
5	1.12

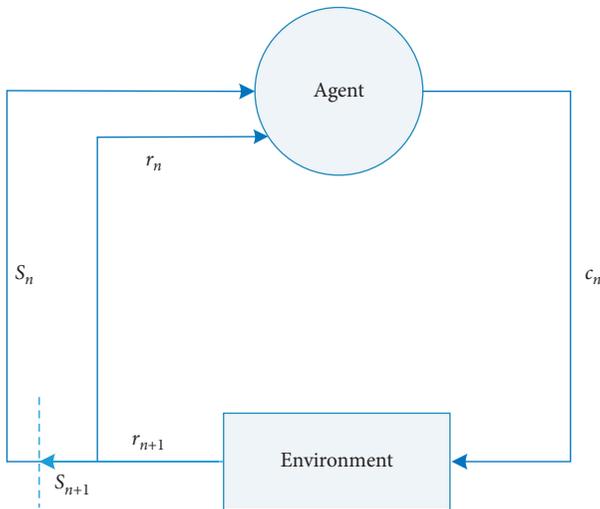


FIGURE 2: Markov decision process.

his grade $S_{n+1}^{(p)}$ for the course under time n . Therefore, write $C_n^{(p)}$ as the action from $S_n^{(p)}$ to $S_{n+1}^{(p)}$, as shown in the following equation. The action $C_{n+1}^{(p)}$ at all times constitutes the action space C :

$$S_n^{(p)} \xrightarrow{C_n^{(p)}} S_{n+1}^{(p)}. \quad (11)$$

- (3) For state transition probability U , the action $C_n^{(p)}$ taken by student p under state $S_n^{(p)}$ is determined by

the timestamp. Once the action $C_n^{(p)}$ is determined, the next state $S_n^{(p)}$ is also determined. At this point, the transition probability between states is determined; that is, $C_n^{(p)} = S_{n+1}^{(p)}$, $U = 1$. The value range of the action $C_n^{(p)}$ is $[1, 5]$.

- (4) For discount factor γ , in the model, each action generates a corresponding reward. However, the same student's viewing time will have different influences on the selection of the next course, and γ is a factor reflecting this influence. The later the reward, the greater the discount, and the return is always limited. Therefore, set $0 \leq \gamma < 1$.
- (5) For reward and punishment function R_{gm} , the punishment and reward function values represent the reward for completing an action in a state. This paper defines the reward and punishment function values R_{gm} as follows:

$$R_{gm}(s_n^{(p)}, c_n^{(p)}) = s_{n+2}^{(p)} - \hat{r}_{px}, \quad (12)$$

where $s_{n+2}^{(p)}$ is the course score of student p when $(n+2)$. px represents the predicted grade of student p for course x calculated using an SVD or SVD++ model. R_{gm} represents the reward and punishment value obtained by student p taking the action $C_n^{(p)}$ under state $S_n^{(p)}$, and the corresponding reward and punishment table can be obtained according to the reward and punishment function.

According to the Markov decision process mentioned above, the action of transferring from one state to the next state corresponds to the score of the course at the next time. Although the name and type of course are ostensibly ignored, students' preferences for the course are implicitly reflected in the timestamp. This process processes the course data set into the form shown in Table 1, where the first number in parentheses reflects the grade given by the students in the corresponding row to the courses in the corresponding column. The second number reflects the timestamp information or chronological order in which

students in the corresponding row watched the courses in the corresponding column. For example, row 1, column 1 (4, 3rd) indicates that student 1 watches course 1 in the third chronological order. So, the timestamp $n = 3$ and student 1 scored 4 for course 1. NaN said the students did not watch the course.

Sort the data in Table 2 by timestamp, and the generated state transition path is as follows:

$$\begin{aligned}
 &4 \longrightarrow 5 \longrightarrow 4 \longrightarrow 2 \\
 &\quad 3 \longrightarrow 3 \\
 &2 \longrightarrow 4 \longrightarrow 5 \longrightarrow 2 \\
 &\quad 4 \longrightarrow 4 \longrightarrow 3 \\
 &3 \longrightarrow 2 \longrightarrow 3 \longrightarrow 5
 \end{aligned} \tag{13}$$

According to Table 2, the rules of the state transfer path are obtained, and the first example is used for illustration. The state transition path $4 \rightarrow 5 \rightarrow 4 \rightarrow 2$ in line 1 reflects that student 1 is watching course 3 when timestamp $n = 1$. Its grade for course 3 is 4. If you look at course 2 at $n = 2$, you give course 2 a grade of 5. If you look at course 1 at $n = 3$, you give course 1 a grade of 4. If you look at course 5 at $n = 4$, you give course 5 a grade of 2. The other four transfer paths were obtained in a similar way.

This state transition path represents the state transition in Markov decision-making process and guides the updating direction of V table.

2.2. Training and Prediction Process. The recommendation algorithm proposed in this paper includes two parts: training and prediction. During the training, SVD++ algorithm is first used for model training on the training set, and SVD++ recommended model is obtained, as shown in formula (3). Then, the reinforcement learning model is trained, and the reward and punishment function shown in (12) is used to calculate the reward and punishment value R_{gm} of state transfer. The reinforcement learning table is updated for the

$$V(s_{n+1}^{(p)}, c_n^{(p)}) = V(s_n^{(p)}, c_n^{(p)}) + \alpha \left[R_{gm}(s_n^{(p)}, c_n^{(p)}) + \gamma wci V(\delta(s_n^{(p)}, c_n^{(p)}), c_n^{(p)}) - V(s_n^{(x)}, c_n^{(x)}) \right], \tag{15}$$

where $V(s_n^{(p)}, c_n^{(p)})$ is a 5×5 V table, $V(s_n^{(p)}, c_n^{(p)})$ starts at 0, $V(s_n^{(p)}, c_n^{(p)})$ is the value of V at the coordinates $(s_n^{(p)}, c_n^{(p)})$, $R_{gm}(s_n^{(p)}, c_n^{(p)}) + \gamma wci V(\delta(s_n^{(p)}, c_n^{(p)}), c_n^{(p)}) - V(s_n^{(x)}, c_n^{(x)})$ is the reward or punishment for choosing the next move, α is learning rate, and γ is discount factor. The higher the V value is, the more the reward you get for performing the next action, and the less the reward you get for performing the next action.

2.2.2. Prediction Process. The prediction process is based on the prediction score obtained by SVD++ recommendation model and combined with V table of training to predict the score of student p on course x . At the same time, it can

TABLE 2: Data evaluation table.

Student	Course 1	Course 2	Course 3	Course 4	Course 5
Student 1	(4, 3rd)	(5, 2nd)	(4, 1st)	NaN	(2, 4th)
Student 2	(3, 1st)	NaN	NaN	NaN	(3, 2nd)
Student 3	NaN	(2, 1st)	(2, 4th)	(5, 3rd)	(4, 2nd)
Student 4	(4, 2nd)	(4, 1st)	NaN	(3, 3rd)	NaN
Student 5	(3, 1st)	NaN	(5, 4th)	(2, 2nd)	(3, 3rd)

optimization model of SVD++ recommendation scoring. During the prediction, the prediction score value is firstly obtained according to SVD++ recommendation model. Then, the optimization model designed in this paper is used to optimize the prediction score, and the final prediction score is obtained. The optimization model designed in this paper is expressed as follows:

$$\hat{r}'_{px} = \hat{r}_{px} + V(s_{n-2}^{(p)}, c_{n-2}^{(p)}), \tag{14}$$

where \hat{r}_{px} is the predicted score of student p for the x course calculated by SVD++ recommendation model. $S_{n-2}^{(p)}$ is the grade of student p watching the course when the timestamp is $(n-2)$ before watching course x . $c_{n-2}^{(p)}$ is the timestamp $(n-2)$ to see the grade of the course. $V(s_{n-2}^{(p)}, c_{n-2}^{(p)})$ is the value of V table in $(S_{n-2}^{(p)}, C_{n-2}^{(p)})$ coordinates. It requires reinforcement learning algorithm and prediction score based on SVD++ recommendation model. It is used to optimize the final predicted score. If the value of $(S_{n-2}^{(p)}, c_{n-2}^{(p)})$ does not exist, the mean value of the current V table is taken. \hat{r}'_{px} is the prediction score after optimization.

2.2.1. Training Process. Firstly, the training set is trained through (3), and the SVD++ recommended model is obtained. Then, the reinforcement learning model was trained. Formula (12) is used to calculate R_{gm} of reward and punishment values, and then R_{gm} is used in the updating process of V value in q-learning Algorithm 1. V table update formula is as follows:

predict the courses that student p has not watched but other students have watched.

In addition, $s_{n-1}^{(p)}$ is not used in the prediction score optimization model shown in (13) for the following reasons. If $s(p)/n - 1$ is used, the optimization model will become $\hat{r}'_{px} = \hat{r}_{px} + V(s_{n-1}^{(p)}, c_{n-1}^{(p)})$. Because $c_{n-1}^{(p)} = s_{n-1}^{(p)}$ and $s_{n-1}^{(p)}$ is the score \hat{r}'_{px} that needs to be predicted, $s_{n-2}^{(p)}$ is chosen.

The data set used in this paper has a default value, that is, there is no graded course information. According to the construction idea of the optimization model in this paper, the scoring information of ungraded courses should be used in the subsequent optimization process. This will lead to the possibility of missing s or c values in $V(s_{n-2}^{(p)}, c_{n-2}^{(p)})$, which will invalidate the optimization model. To avoid this

```

Input:
  The number of students N, the courses that students have graded W, the learning rate  $\alpha$ , and the discount factor  $\gamma$ 
Output:
   $V(s, c)$ 
S1: Initialize  $V(s, c) = 0$  for any  $s \in S, c \in C$ 
S2: Training SVD++ model, calculate  $\hat{r}_{px}$  from equation (3)
S3: Get the initial state S and action A from the data.
S4: For each episode  $x = 1: T$  do
  For  $z = 1: Wx$  do
    S5: Calculate the reward and punishment functions according to equation (8).
    S6: Apply the calculated punishment and reward functions to equation (10)
    S7: Update the V table.
  end for
end for

```

ALGORITHM 1: The pseudocode of proposed algorithm training process.

situation, SVD++ model is adopted in this paper to predict missing values and then fill them to solve the problem of sparse data.

In addition, when $n = 1, 2$, the boundary $s_{n-2}^{(p)}$ and $c_{n-2}^{(p)}$ will exceed the subscript range, and there will be no corresponding value. Therefore, the prediction score of the last two columns is used as the prediction score data of column -1 and column 0 in this paper to ensure data integrity.

3. Experiment and Analysis

3.1. Experimental Data Set. To verify the actual effect of the algorithm in this paper on the recommendation model of ideological and political courses in colleges and universities, sklearn library based on *Python* 3.6.5 kernel provided by GitHub open source platform is adopted. The main library packages used are numpy-1.14.3, NetworkX-2.3, Keras-2.0.5, and sklearn-0.20.0. The experiment was conducted in ubuntu-16.04 environment. Multidimensional constraints such as scale, feature, target, and noise are set by programming, and sklearn library is used to generate simulated data sets that meet the conditions. The simulated data set of the experiment in this paper contains information of 570 ideological and political education courses, information of 13,000 college students, and information of 50,000 college students' operation and grading of courses.

3.2. Performance Indexes. To evaluate the recommended performance of the proposed algorithm, MAE and RMSE were used as evaluation indexes.

MAE represents the average absolute error between the predicted value and the true value. The smaller the MAE is, the higher the recommendation accuracy is. It is defined as follows:

$$M_{MAE} = \frac{\sum_{x,y} |\hat{r}_{xy} - r_{xy}|}{t} \quad (16)$$

RMSE represents the square root of the sum of the squares of the deviation between the predicted value and the true value and the ratio of the predicted number t . RMSE

reflects the dispersion degree of samples, and the smaller the RMSE, the higher the recommendation accuracy. It is defined as follows:

$$R_{RMSE} = \sqrt{\frac{\sum_{x,y} (\hat{r}_{xy} - r_{xy})^2}{t}}, \quad (17)$$

where \hat{r}_{xy} is the predicted score value, r_{xy} is the true value, and t is the number of predicted scores.

3.3. Feasibility Analysis. Since the learning rate α and discount factor γ can be adjusted dynamically, it is necessary to study the influence of the changes of α and γ in the algorithm in this paper on the prediction performance. The experimental results are shown in Figure 3. As can be seen from Figure 3, when α is constant, γ increases from 0.4 to 0.6, and RMSE of the proposed algorithm decreases continuously. The results were best when $\alpha = 0.000003$ and $\gamma = 0.6$. At this time, RMSE can reach 0.81948, which is 0.0086 lower than before. Thus, the feasibility of the proposed algorithm is proved. This configuration is used for comparison in subsequent experiments.

3.4. Comparison Experiment. To verify the effectiveness of the algorithm in this paper, representative matrix decomposition algorithm [23], direct trust matrix decomposition algorithm [24], and indirect social relationship matrix decomposition algorithm [25] were selected for comparison in the experiment. All experiments were carried out for 10 times, and the experimental results were averaged.

Tables 3 and 4 show the error values of different algorithms. By observing the data in the table, it can be seen that the error value of the algorithm in this paper is lower than that of other comparison algorithms, and the error comparison becomes more obvious with the increase of training sets. According to Tables 3 and 4, the more the training sets, the lower the error rate of the algorithm. Compared with the algorithm [23], the algorithm [24] reduces the error value due to the introduction of direct social relations. On the basis of the algorithm [24], the algorithm [25] established

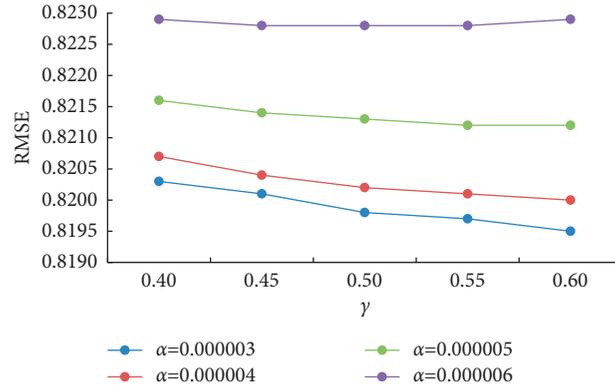


FIGURE 3: Influence of γ variation on RMSE at constant α .

TABLE 3: Comparison of error values with MAE.

Training set (%)	Index	Paper [23]	Paper [24]	Paper [25]	Proposed
40	MAE	1.0207	0.9841	0.8656	0.8324
60	MAE	1.0179	0.9588	0.8492	0.7810
80	MAE	1.0069	0.9406	0.8354	0.7500

TABLE 4: Comparison of error values with RMSE.

Training set (5%)	Index	Paper [23]	Paper [24]	Paper [25]	Proposed
40	RMSE	1.3489	1.2718	1.1151	1.1024
60	RMSE	1.3443	1.2429	1.1333	1.0663
80	RMSE	1.3223	1.2066	1.1890	1.0029

indirect social relations by considering the degree of influence between each student. This relationship includes both direct and indirect connections between students, so the error value is further reduced. However, these algorithms only analyze the degree of influence among students, ignoring the potential connection between courses and the difference of influence degree of different neighbours on different nodes. The algorithm in this paper constructs preference vector and feature vector, respectively, according to the influence degree of scoring and behaviour from different perspectives of students and courses. At the same time, considering the time effect of student scoring, the recommendation model is transformed into Markov decision process. Therefore, the proposed algorithm has a higher recommendation rate than other algorithms.

To verify whether the algorithm in this paper can reduce the iterative convergence times, the original data set is taken as the training set, and the experimental results are shown in Figure 4. It can be seen that compared with other comparison algorithms, the iteration times of the algorithm in this paper quickly reach convergence. Moreover, the recommendation efficiency of the proposed algorithm is significantly higher than other algorithms. This is because the algorithm in this paper fully mines the potential connections between them from the perspectives of students and courses, so it not only improves the recommendation efficiency but also reduces the convergence times.

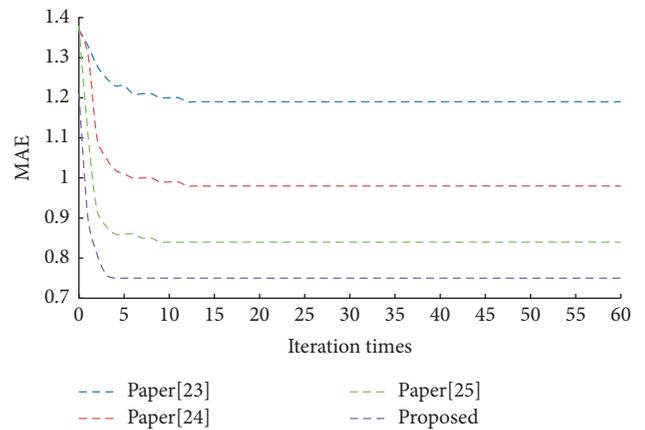


FIGURE 4: Comparison of iterative convergence times.

At the beginning of iteration, the error value of comparison algorithm is about 1.38. By observing Figure 4, it can be seen that the error value of the zeroth iteration of the algorithm in this paper is lower than 1.38. As the number of iterations increases, the algorithm [23] achieves convergence and maximum error only after 20 iterations. However, the algorithm in [24] and the algorithm [25] indirectly corrected the preference vector of students, and reached convergence after 15 iterations, followed by recommendation efficiency.

Because the algorithm in this paper optimizes the preference vector and feature vector from the two aspects of students and courses, respectively, convergence has been achieved and the error value is small at the fourth time. Experimental results further verify the proposed algorithm's recommendation accuracy and convergence performance.

4. Conclusion

Ideological and political education curriculum runs through the whole process of higher education teaching and is an essential system for universities to cultivate high-quality talents. To realize personalized recommendations of ideological and political education courses, this paper proposes a recommendation algorithm based on the analytic hierarchy process and an improved collaborative filtering algorithm. The action of students scoring courses under different time stamps is transformed into Markov decision process. Through analytic hierarchy process, the student behavior data is converted into course preference value. The collaborative filtering algorithm is combined with the reinforcement learning reward and punishment process, and the prediction effect was improved by adjusting the influence factors. Experimental results show that the feasibility of the proposed system is verified, and the system has good accuracy and convergence performance. In the case of insufficient training set data, how to improve the accuracy of the algorithm is the future research work.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] G. Gong, "The Integration of Ideological and Political Education in Colleges and Universities and Employment of College Students," *DEStech Transactions on Social Science, Education and Human Science*, 2017, <https://www.dpi-proceedings.com/index.php/dtssehs/article/view/13031>.
- [2] J. Zhang, X. Hu, Z. Ning et al., "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2633–2645, 2018.
- [3] L. G. Xing, I. A. Abiodun, C. W. Khuen, and B. T. Teik, "A Personalized Recommendation Framework with User Trajectory Analysis Applied in Location-Based Social Network (LBSN)," in *Proceedings of the IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS)*, pp. 1–6, IEEE, Bangkok, Thailand, August 2017.
- [4] X. Yang, Z. Huan, Y. Zhai, and T. Lin, "Research of personalized recommendation technology based on knowledge graphs," *Applied Sciences*, vol. 11, no. 15, p. 7104, 2021.
- [5] D. Xu, "Research on music culture personalized recommendation based on factor decomposition machine," *Personal and Ubiquitous Computing*, vol. 24, no. 2, pp. 247–257, 2020.
- [6] R. Alaa, M. Gawich, and M. Fernández-Veiga, "Personalized Recommendation for Online Retail Applications Based on Ontology Evolution," in *Proceedings of the 2020 6th International Conference on Computer and Technology Applications*, pp. 12–16, Bogotá, Colombia, April 2020.
- [7] W. Zhu, W. Liu, and Z. Liu, "News personalized recommendation based on event ontology," *Computer Engineering*, vol. 45, no. 6, pp. 267–272, 2019.
- [8] H. Zarzour, Z. Al-Sharif, M. Al-Ayyoub, and Y. Jararweh, "A New Collaborative Filtering Recommendation Algorithm Based on Dimensionality Reduction and Clustering techniques," in *Proceedings of the 9th International Conference on Information and Communication Systems (ICICS)*, pp. 102–106, IEEE, Irbid, Jordan, April 2018.
- [9] Z. Cui, X. Xu, F. Xue et al., "Personalized recommendation system based on collaborative filtering for IoT scenarios," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 685–695, 2020.
- [10] X. Hu, J. Cheng, M. Zhou et al., "Emotion-Aware cognitive system in multi-channel cognitive radio ad hoc networks," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 180–187, 2018.
- [11] A. Suglia, C. Greco, and C. Musto, "A Deep Architecture for Content-Based Recommendations Exploiting Recurrent Neural networks," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pp. 202–211, Bratislava, Slovakia, July 2017.
- [12] M. Oppermann, R. Kincaid, and T. Munzner, "Vizcom-mender: computing text-based similarity in visualization repositories for content-based recommendations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 495–505, 2020.
- [13] X. Wang, X. He, and Y. Cao, "Kgat: knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 950–958, Anchorage AK, USA, August 2019.
- [14] M. Naumov, D. Mudigere, H. J. M. Shi, and J. Huang: Deep Learning Recommendation Model for Personalization and Recommendation Systems, 2019.
- [15] R. Mu, "A survey of recommender systems based on deep learning," *Ieee Access*, vol. 6, pp. 69009–69022, 2018.
- [16] G. Manogaran, R. Varatharajan, and M. K. Priyan, "Hybrid recommendation system for heart disease diagnosis based on multiple kernel learning with adaptive neuro-fuzzy inference system," *Multimedia Tools and Applications*, vol. 77, no. 4, pp. 4379–4399, 2018.
- [17] Z. Ji, H. Pi, W. Wei, B. Xiong, M. Wozniak, and R. Damasevicius, "Recommendation based on review texts and social communities: a Hybrid model," *IEEE Access*, vol. 7, Article ID 40416, 2019.
- [18] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.
- [19] X. Wu, X. Yuan, C. Duan, and J. Wu, "A novel collaborative filtering algorithm of machine learning by integrating restricted Boltzmann machine and trust information," *Neural Computing & Applications*, vol. 31, no. 9, pp. 4685–4692, 2019.
- [20] R. Ahuja, A. Solanki, and A. Nayyar, "Movie Recommender System Using K-Means Clustering and K-Nearest Neighbor," in *Proceedings of the 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 263–268, IEEE, Noida, India, January 2019.

- [21] C. Bakir, “Collaborative filtering with temporal dynamics with using singular value decomposition,” *Tehnički Vjesnik*, vol. 25, no. 1, pp. 130–135, 2018.
- [22] R. Kumar, B. K. Verma, and S. S. Rastogi, “Social popularity based SVD++ recommender system,” *International Journal of Computer Application*, vol. 87, no. 14, 2014.
- [23] C. Chen, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, “Efficient neural matrix factorization without sampling for recommendation,” *ACM Transactions on Information Systems*, vol. 38, no. 2, pp. 1–28, 2020.
- [24] G. Guo, J. Zhang, and N. Yorke-Smith, “Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, Austin, TX, USA, January 2015.
- [25] T.-w. Zhang, W.-p. Li, L. Wang, and J. Yang, “Social recommendation algorithm based on stochastic gradient matrix decomposition in social network,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 601–608, 2020.