Design of a Personalized Recommendation System for Learning Resources based on Collaborative Filtering

Mingxia Zhong, Rongtao Ding* School of E-commerce, Zhejiang Business College, Hangzhou 310053, China *Email: dingrongtao_zvcc@126.com

Received: June 16, 2021. Revised: December 6, 2021. Accepted: January 5, 2022. Published: January 7, 2022.

Abstract—At present, personalized recommendation system has become an indispensable technology in the fields of e-commerce, social network and news recommendation. However, the development of personalized recommendation system in the field of education and teaching is relatively slow with lack of corresponding application.In the era of Internet Plus, many colleges have adopted online learning platforms amidst the coronavirus (COVID-19) epidemic. Overwhelmed with online learning tasks, many college students are overload by learning resources and unable to keep orientation in learning. It is difficult for them to access interested learning resources accurately and efficiently. Therefore, the personalized recommendation of learning resources has become a research hotspot. This paper focuses on how to develop an effective personalized recommendation system for teaching resources and improve the accuracy of recommendation. Based on the data on learning behaviors of the online learning platform of our university, the authors explored the classic cold start problem of the popular collaborative filtering algorithm, and improved the algorithm based on the data features of the platform. Specifically, the data on learning behaviors were extracted and screened by knowledge graph. The screened data were combined with the collaborative filtering algorithm to recommend learning resources. Experimental results show that the improved algorithm effectively solved the loss of orientation in learning, and the similarity and accuracy of recommended learning resources surpassed 90%. Our algorithm can fully satisfy the personalized needs of students, and provide a reference solution to the personalized education service of intelligent online learning platforms.

Keywords—online learning platform; knowledge graph; collaborative filtering algorithm; personalized recommendation system for learning resources.

I. INTRODUCTION

The coronavirus (COVID-19) epidemic enhances the importance of digital online learning to college education, which is already a popular learning method in the era of Internet Plus. Many universities and vocational colleges have developed their own learning resource libraries and online course learning platforms. However, the massive learning resources on these platforms often overload the college students, making them difficult to keep orientation in learning.

Lost in the sea of information, the students face a growing difficulty in finding the necessary information accurately and efficiently. This gives rise to a novel technology called personalized education service. To support the digital personalized learning, it is critical to develop a personalized recommendation system of learning resources that caters to the different needs of various learners.

In recent years, there have been many literature reports on Personalized Teaching Resource Recommendation System. Tian Feng [1] had proposed a personalized learner model construction method to solve the problem of incomplete description of learner characteristics in the current teaching resource recommendation system. Zhao Xuekong [2] had proposed a hybrid recommendation model of educational resources for users' individual needs and how to build a hybrid intelligent recommendation scheme of educational resources based on content and collaborative filtering in the Internet environment. Xu Yajun [3] had proposed a Personalized Learning Resource Recommendation Algorithm for K12 learning platform. Experimental data show that the resource accuracy of the platform is more than 90%, but the algorithm does not consider the user tag and the possible cold start problem of user based collaborative filtering algorithm. On the basis of their research results, combined with the specific situation of our college students and network teaching platform, this paper expounds the related technologies of the personalized education resource recommendation system, and focuses on the system recommendation process and technical architecture, the construction of knowledge map, user similarity, collaborative filtering, evaluation index and other

contents. The research goal of this paper is to build an effective personalized recommendation system of teaching resources by using knowledge mapping technology, analyze the classic cold start problem in personalized recommendation system, and use hybrid recommendation technology to improve the accuracy of recommendation.

An intelligent personalized recommendation system for learning resources should meet the following goals: First, when a student logs in the learning platform, the system needs to push the interested learning contents to him/her, according to his/her learning habits, behavior patterns and points of interest; Second, the system needs to search for the students sharing similar interests with he/she, using user similarity algorithm, and recommend their interested courses or learning resources to him/her.

The most mature and popular algorithm for the search of interested learning resources is the collaborative filtering algorithm in many e-commerce recommendation systems [4, 5]. Take the recommendation of online learning resources for the professional online courses on e-commerce for example. The score matrix of these courses is displayed in Table 1, where each row represents a student, and each column represents a

course. Every element in the matrix is the score of a course rated by a student (the full mark is 10 points). The greater the score, the more the student prefers the course.

By analyzing the matrix data, the recommendation algorithm of the system notices that the three students rated the same scores for each of the last three courses. Thus, the system will recommend *Visual Design in E-Commerce* to student C, because this course received high ratings from students A and B, who have similar preferences as student C.

Nevertheless, the recommendation algorithm cannot make effective recommendations if the courses and users are new to the system. Taking the score matrix in Table 2 for instance, the recommendation algorithm cannot find students with similar preferences as the new student E, due to the lack of his/her scores. Hence, the online learning platform will be unable to push relevant courses to the student. Similarly, the platform cannot make personalized recommendations for the new course *New Media Marketing*, due to the lack of scores or user data on the course. This is the classic cold start problem [6] of recommendation algorithms.

Table 1. Score matrix of professional online courses on e-commerce

	1								
	Visual Design in E-Commerce	Online Store Decoration	Visual Marketing of Online Stores	Fundamentals of E-Commerce					
Student A	9	7	8	5					
Student B	9	7	8	5					
Student C		7	8	5					
Student D	5	4	4	4					

Tuble 2. Secte matrix of professional online courses on e commerce with a new stadent								
	Visual Design in	Online Store	Visual Marketing of Online	Fundamentals of	New Media			
	E-Commerce	Decoration	Stores	E-Commerce	Marketing			
Student A	9	7	8	5				
Student B	9	7	8	5				
Student C		7	8	5				
Student D	5	4	4	4				
Student E								

Table 2. Score matrix of professional online courses on e-commerce with a new student

To solve the cold start problem in learning resource recommendation, the authors summarized the features of the online learning platform in our university, clarified the knowledge structure of the learning resources on the platform, and extracted the data on learning behaviors of the platform. On this basis, a knowledge graph was prepared to represent the knowledge points in the learning resources of the platform, and used to improve the collaborative filtering algorithm for personalized recommendation of online learning resources to college students.

II. KNOWLEDGE POINT REPRESENTATION

From the traditional information age to the current era of big data, many intelligent apps have emerged in the field of e-commerce, namely, the Qianrenqianmian (literally, "a thousand faces for a thousand people") recommendation algorithm of Taobao. All of the personalized intelligent recommendation systems of major platforms need to collect, store, and model complex data on user behavior [7], which cover the user trajectories in multiple fields, and thus form an ecosystem with online e-commerce platform at the core. The intelligent algorithms promote the value creation from the user data.

In the education field, a large amount of data (e.g., courseware, videos, exercises, and question banks) have also been accumulated on online learning platforms through the digitization of traditional learning resources. These unstructured and diverse data cannot be directly subject to data mining or machine learning, if the learning resources are recommended personally like e-commerce products. Therefore, a suitable representation technology is the key to unified processing of online learning resources and development of intelligent application algorithms [8].

This paper adopts the knowledge point representation technique of learning resources. Most course contents are unfolded around some knowledge points. Each knowledge point can be represented as a node in a preset knowledge graph, using graph network technology. Then, the characterization vector thus obtained is taken as the initial input of the knowledge point, and used for further learning through model training. This representation of knowledge point applies to different tasks, because of its good mobility and retainment of the relationship between knowledge points [9]. The knowledge graph of *Operation and Promotion of Online Stores* is shown in Figure 1.

In this paper, the knowledge points of each course are treated as the entities in the knowledge graph, and their relationships are viewed as edges to generate recommendations. This practice ensures the recommendation accuracy, and makes the recommendation results explainable. Because most of the course teaching content is usually around some knowledge points, taking the knowledge points of each course as the entity of the knowledge graph can ensure that the recommendation results obtained by the recommendation algorithm are compact knowledge points or courses associated with knowledge points; second, whether the recommendation results are reasonable and whether they are really similar interest points of students can be based on the knowledge graph Whether the spectrum contains a certain knowledge point to explain. Figure 2 presents the entity relationships in a knowledge graph. Taking course recommendation for example, the knowledge graph consists of entities (students, courses, knowledge points, and teachers) and relationships (interaction, learning, belonging, and teaching).

The basic unit of knowledge graph consists of two parts: (1) "entity- relation -entity" triplet; (2) entity and its corresponding attributes are usually value pairs or (key, value) relation expressions in programming languages. As shown in Figure 1 and Figure 2, the network knowledge base structure will be transformed into a structured language that can be understood by computer. For example, about teaching resource platform, entity relationship triples can be "teacher, teaching, course" and "student, major, computer science and technology". For example, if the knowledge point of a programming language course is regarded as the entity of the knowledge map, then the knowledge point triple of a specific elaboration data of the course can be expressed as <data type, dictionary, key / value>. In the previous step, the unstructured data resources have been preprocessed into structured data that can be easily recognized by computer. Next, we consider using graph database to store. Because graph database is much more flexible than relational database, when the content of knowledge graph increases and the system level becomes more, the efficiency of graph database will be hundreds or even thousands of times higher than that of relational database. In this paper, Neoj graphic database is used to store structured data on the network to form a graph instead of a table. Neoj graph database contains a high-performance graph engine, and has the important

characteristics of mature database, such as transaction, index and so on. Taking a learning video of a course as an example, the steps of storing the relationship among teachers, courses and learning resources in Neoj database are as follows:

(1) Create a course node, including node name, tag name, attribute name, attribute value, etc. For example, a course ID is 1470679, the course name is "Python language and data analysis", and the subject is business data analysis, with 48 lesson periods and 3 credits. The course node can be described as follows:

create (c:Course{id: 1470679, name:" Python language and data analysis ", lesson period:48, credit:3})

(2) Increase the teaching relationship between teachers and courses, For example, create a 45 years old female teacher named Andy who teaches this course. The teacher node can be described as follows:

create(t:Teacher {id:2019000121,name:"Andy",age:45,sex:" F",teach:" Python language and data analysis "})return t

(3) The learning resources of the course may include documents, audio, video, test questions, etc. These resources need to be associated with the course. For example, take video as an example to create related video nodes, including video name, subordinate course, lecturer, video duration, video format, etc. The video node can be described as follows:

create (v:Video{id: 1470679001, name:" Data Type", related course: " Python language and data analysis ", lecturer: "Tom", video duration:15, video format: "MP4"})

Then we use the same second step method to associate the course video with the course. Through the above steps, all the resources, students, teachers and other entities on the teaching platform are associated and stored in the graph database to form a graph data network. Of course, the construction of the knowledge graph structure of the whole course is not a one-time generation, it needs to be repeated and updated to accumulate slowly. The first is data collection: the resources on the teaching platform are mainly unstructured data and semi-structured data, which need to be stored and processed into structured data by Neo4j graph database; the second is information extraction: the three contents of attribute, relationship and entity are extracted from the content data source, which can become the feature vectors of the later algorithm input.

In the same principle, user similarity can be expressed by user portraits, and user portraits can also be constructed by knowledge graphing technology. For example, the triple entity relationship of students can be expressed by < student a, classmate, student b >, < student a, gender, male > and other methods, and the knowledge graphing information can be stored in the Neoj map database.



Fig. 1. Knowledge graph of Operation and Promotion of Online Stores



Fig. 2. Ontologies and relationships in the knowledge graph

III. DESIGN OF SYSTEM FRAMEWORK

Our personalized recommendation system for learning resources consists of five modules: a data layer, a preprocessing layer, an algorithm layer, a service layer, and an application layer (Figure 3).

(1) Data layer

A user learning model was constructed by analyzing the knowledge graphs of the courses in the course center, as well as the information on user behaviors (e.g., learning paths) on the learning platform.

To ensure the stability and accuracy of the data, this paper chooses the SDK (Software Development Kit) data collection method [10, 11]. Firstly, the sensors were embedded in the learning platform and mobile learning app. Specifically, invisible probes were buried in the page to collect the data on learning behaviors, operation logs, and stored data. Next, the server relocated the collected log data to the storage device. Finally, features were extracted from the data, and imported to the preprocessing layer.

(2) Preprocessing layer

Based on the collaborative filtering recommendation algorithm, the data input model was established through the representation of knowledge points and the creation of user portraits. Relying on the aforementioned knowledge graph-based method, the knowledge point representation made the entities corresponding to the knowledge graph the input of the subsequent algorithm. Meanwhile, each user portrait was a model of the learning interests, learning behaviors, and attributes of each student.

After surveying the students' points of interest, the learning

behaviors of the students were analyzed, and combined with the individual needs of learning to divide the students into different groups. Then, the typical features of each group were extracted, and recorded as structured information, producing a summary of user features. On this basis, the eigenvalues of learning behaviors could be calculated for users who access the platform.

The collected user behavior data were stored in the database. In this paper, the original data are processed through extract, transform, and load (ETL: Extract-Transform-Load) method [12]. To make the stored data more suitable for model training, the ETL tool was employed to eliminate the invalid duplicate data, i.e., the repeated data with no effect on user behaviors, and convert unstructured and semi-structured data into structured data. This processing is very necessary, because the SDK often captures lots of useless data, such as blanks, special symbols, or even meaningless types of data. These data will increase the computing load of subsequent algorithm, and hinder the real-time data processing in the system.

(3) Algorithm layer

The algorithm layer mainly completes the data modeling in the entire system. Here, the vector space model (VSM) [13, 14] is adopted to describe the features of users and ontologies. Each dimension of the vector represents the data on a feature or an ontology of a knowledge point.

The first step is to establish a VSM. From the massive preprocessed datasets, a set of n keyword features was obtained as $(t_1, weight_1), (t_2, weight_2), \dots, (t_n, weight_n)$, where t_{1-n} are the keywords, and weight_{1-n} are the weights of these keywords, and used to set up an n-dimensional eigenvector space.

Next, the model was trained by common methods, such as course-based collaborative filtering, user-based collaborative filtering, and knowledge-based recommendation. Finally, the trained model data were outputted.

(4) Service layer

This layer is mainly responsible for recommendation engine and learning behavior monitoring of the system. Considering the computing load of the engine, a recommendation model was created to recommend learning resources based on the relationships between knowledge points and the training model. In fact, in the current recommendation system, few recommendation engines only use one recommendation strategy. Generally, different recommendation strategies are used in different scenarios to achieve the best recommendation effect. For example, Amazon recommendation will be based on the recommendation of users' historical purchase data and the recommendation of the items currently viewed by users, And the popular articles based on the popular preferences are recommended to users in different areas, so that users can find their real interests from all-round recommendations. The recommended technology of teaching resources in this paper also needs to adopt hybrid recommendation technology to meet the personalized needs of users.

(5) Application layer

The personalized learning contents or courses recommended by the system are displayed on the client page, waiting to be selected by students. Here is mainly related to the sorting algorithm, sorting is fast and accurate, fast refers to the efficient feedback results, accurate refers to the high accuracy of the recommended results. Specifically, after generating candidate objects, the generated candidate objects are scored and sorted to get the final item list to be pushed. The evaluation of ranking results mainly includes two steps: the first step is offline evaluation. Data validation is performed on the offline prepared validation set. If the effect is better than the previous model data, the second step is performed. The effect here includes multiple indicators, such as accuracy, coverage, diversity, etc. The second step: online experiment, directly for online users to carry out AB bucket experiment, first small flow experiment for a period of time, generally in about 10 days to half a month, compared with the experimental group and the control group data; If the experimental data is positive, more users will be pushed; If the data continues to be stable, all users will be pushed.



Fig. 3. Framework design and data flows of our system

IV. USER-BASED COLLABORATIVE FILTERING

Our collaborative filtering recommendation algorithm mainly operates based on the logs of the historical behaviors of students on the online learning platform of our university: the students with similar learning interests as the target student was calculated through mathematical modeling; the knowledge points or courses that are potentially needed by the target student were determined according to logs of the historical behaviors of these similar students, and recommended to the target student.

The main steps of our algorithm are as follows:

Step 1. Data preprocessing

Collect and store the data on learning behaviors and user portraits of the students, the resources filtered by knowledge graph are taken as the samples of collaborative filtering, and the recommended entities are resources. The watching, liking and collecting of resources are taken as the user's rating of resources, the knowledge point attribute of comprehensive resources, classmates and subects are taken as different weights, and the score is calculated as the user's rating of resources

Step 2. Calculation of user similarity

Select Spearman's rank correlation coefficient to measure the similarity between users through comparative experiments.

Step 3. Course recommendation

Find the n users with the highest similarity as the target student through collaborative filtering, and recommend suitable courses to the target student.

A. Data preprocessing

(1) Student evaluation index

Our collaborative filtering recommendation algorithm takes the resource scores rated by the students as the basis for recommendation. The comprehensive evaluation index $score_{evaluate}$ of a course depends on multiple factors: the score $score_{grade}$ rated by the students (full mark: 10 points); whether it is liked *isLiked*; whether it is collected *isCollected*; the percentage of view of the course to all courses:

$$score_{evaluate} = score_{grade} / 10 + w_{collect} \times isCollected$$

+ $w_{like} \times isLiked + w_{view} \times percentOfView$

where, w is the weight of each behavior; *isLiked* and *isCollected* are Boolean data (their values are 1 or 0, depending on whether the course is liked or collected by the

target student); *percentOfView* is a floating-point number.

(2) Knowledge point index

The teaching platform generally has the characteristics of periodic teaching. In order to achieve more accurate recommendation, the buried point we set will record the date when the student enters the platform, and then calculate the knowledge points that the student learns on that day according to the date. In order to facilitate the calculation of the later algorithm, this paper divides the time of one year and calculates the radius of each day in one year. The time values are calculated according to the months: January to February, winter vacation, no interval, unified as 0; march to June, spring semester, the interval is (0, 0.5), the value is calculated according to the number of days in a year, reserved to three decimal places; July to August, summer vacation, no interval, unified as 0.5; September to December, autumn semester, the interval is (0.5, 1), the value is calculated according to the number of days in a year, to three decimal places.

At the same time, in order to facilitate the calculation, this paper takes the knowledge points of one academic year as 0 - 1.0, and each knowledge point gets the corresponding radius value. Through the radius of each day and the radius value of knowledge points, we can calculate the corresponding knowledge points of each day. In this paper, we take the three nearest knowledge points corresponding to the date radius value. Through the knowledge points and users, the corresponding entity resources are obtained from the knowledge map, including the resources of the latest knowledge points, the resources watched by users, the resources learned by classmates, the resources learned by classmates, and the resources are used as the input of collaborative filtering.

The popularity of watching resources on teaching platform is closely related to learning progress. This paper takes the resources of three knowledge points near the date radius value, and calculates the knowledge point index

$$score_{kp} = 1 - |r_{date} - r_{kp}|$$
 $score_{kp} = 1 - |r_{date} - r_{kp}|$

where, r_{date} represents the date radius, r_{kp} represents the knowledge point radius.

(3) Cooperative index

Generally speaking, the teaching progress of students of the same grade or subject in a school is basically similar. Each student's subject and grade are stored in the Neoj graph database. Therefore, this paper takes the learning resources of classmates and students of the same major as the synergy index, and the formula is as follows:

$$score_{collaborative} = \begin{cases} w_{class} \times f_{class} \\ w_{subject} \times f_{subject} \end{cases}$$

where, w represents class or subject weight, f represents classmate or not, Boolean type, value is 1 or 0.

Finally, based on the above three indexes, the students' rating formula for resources is as follows:

 $score = W_{evaluate} \times score_{evaluate} + W_{kp} \times score_{kp} + W_{collaborative} \times score_{collaborative}$

where, *score* is the value of each index , w the weight of each index.

B. Calculation of user similarity

User similarity can be measured by correlation coefficients in the field of statistics. The most common correlation coefficients are Euclidean distance, Pearson correlation coefficient, and Spearman's rank correlation coefficient. Euclidean distance is a yardstick of the distance between two points in space. If similarity is defined by Euclidean distance, it is impossible to reflect the effect of overlapped data on similarity [15]. Compared with Pearson correlation coefficient, Spearman's rank correlation coefficient has relatively relaxed requirements on data conditions. This coefficient works well even if the data contain outliers. Since the preprocessed sample data of the learning platform are either unstructured or semi-structured, the Spearman's rank correlation coefficient was chosen to measure user similarity through comparative experiments.

Spearman's rank correlation coefficient estimates the correlation between two variables X and Y. The two variables can be viewed as two sets, each of which has N elements. The i-th $(1 \le i \le N)$ values of the two stochastic variables can be represented by X_i and Y_i , respectively. Sorting X and Y (in descending or ascending order), the element ranking sets x and y were obtained for the two variables. Then, X_i and Y_i reflect the ranking of X_i in X and that of Y_i in Y, respectively. Next, a ranking difference set $d_i = x_i - y_i$ ($i(1 \le i \le N)$) was obtained by subtracting each element in set x with the corresponding element in set y. The Spearman's rank correlation coefficient between the two stochastic variables X and Y can be derived from X, Y, or d [16]:

$$\rho = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{N(N^2 - 1)}$$

C. Course recommendation

In the above step, the K students, whose learning interests are the most similar with target student u, were identified. The data on the K students were eventually fed back in the form of user dictionary S(u, K). Then, user-based collaborative filtering (UserCF) was called to recommend courses to the target student based on the data of the K students. The interest of user u in course i can be measured by:

$$p(u,i) = \sum_{v \in S(u,K) \cap N(i)} W_{uv} r_{vi}$$

where, S(u, K) is the set of the K students; N(i) is the set of users that have learned course i; w_{uv} is the similarity between users u and v in learning interest measured by Spearman's rank correlation coefficient; r_{vi} is the interest of user v in course i.

The user-based collaborative filtering eventually outputs a dictionary to be recommended. The courses in the dictionary were sorted in descending order by the calculated interest p(u,i), and then pushed to the client of the target student.

The specific program flow and work content of the UserCF algorithm are shown in the following figure 4:

V.HYBRID RECOMMENDATION TECHNOLOGY

But the biggest drawback of collaborative filtering algorithm mentioned above is the cold start problem. When new users enter the platform, there is no historical information, so the system often can't perceive its existence, and it is difficult to make personalized recommendation. In front of us, we use the construction of knowledge graph to express the course content and knowledge system. At the same time, we use the information storage technology of Neoj graph database to label each entity and establish the entity relationship between them. Therefore, it is very conducive to the operation of another algorithm content-based recommendation algorithm.

The content-based recommendations (CBR) is based on the relevant information of learning resources, students' relevant information and students' operation behavior of learning resources to build a recommendation algorithm model to provide recommendation services for students. Content based recommendation algorithm generally only depends on the user's own behavior to provide recommendations for users, and does not involve the behavior of other users, so it does not have the problem that new users lack historical information in cold start. We just need to start the collaborative learning algorithm based on user's content to solve the problem of user's similarity.



Fig. 4. Process and content of UerCF algorithm

Therefore, this paper uses a hybrid recommendation technology which combines user based collaborative filtering algorithm and content-based recommendation algorithm, and the weighted hybrid recommendation technology can get better recommendation results, that is, first run the two recommendation algorithms independently to generate some candidate results, and then combined weighted, and finally get the final recommendation ranking results. A relatively simple weighting method can give the same weight value to the collaborative filtering recommendation results and the content-based recommendation results, and then generate the recommendation results. However, the fixed weighting system does not consider that the quality of the algorithm will change under different learning objects and learning scenarios, so it considers setting training samples, comparing the user's evaluation of the recommendation results and whether it is consistent with the system's prediction, generating weighted model according to the training results, and dynamically adjusting the weight to generate the recommendation results.

The implementation steps of CBR algorithm are as follows: Step1: Compute the similarity of learning resources

In Neoj database, we have stored the entity content of structured information, such as video, including video name, course, lecturer, video duration, video format, knowledge point label, score and other dimensional data. For this kind of data, we can take a field (also a feature) as a dimension of the vector. The each dimension may be not a numerical value, but the form is still vectorized. That is, the vector space model (VSM) was mentioned in the algorithm layer description in the third section in the article. We can calculate the similarity between the two videos in the following way.

Suppose that two video resource vectors are represented as:

 $v_1 = (p_1, p_2, p_3, \dots, p_k)$, $v_2 = (q_1, q_2, q_3, \dots, q_k)$

In this case, the similarity between the two videos can be expressed as follows:

$$sim(V_1, V_2) = \sum_{t=1}^k sim(p_t, q_t)$$

where, $sim(p_t, q_t)$ represents the similarity between two components of the vector. The similarity between two components can be calculated by using the similarity algorithm formula mentioned above like Spearman's rank correlation coefficient.

Step2: Compute users' preference for learning resources

User 's preference for learning resources sim(u, s) can be calculated by the following formula, where U is the list of learning resources operated by all users, and $score(s_i)$ is user u 's preference for learning resources like the evaluation index mentioned in Section 4.1 above, $sim(s_i, s)$ is the similarity between learning resources and s.

$$sim(u, s) = \sum_{s_i \in U} score(s_i) \times sim(s_i, s)$$

In this way, the user's preference for learning resources is calculated, and then sorted in descending order based on the value of preference, so that the TopN mentioned later can be used to recommend to other users.

VI. EXPERIMENTAL RESULTS

Due to the diversity of learning materials on the teaching platform, the data structure is complex. Although we mentioned the use of knowledge mapping to do structured processing, which greatly reduces the amount of data calculation, the complexity and diversity of data structure still make a recommendation system face huge data calculation. In order to test the advantages and disadvantages of the algorithm faster and increase the real-time performance of the recommendation system, this paper adopts the offline experimental method, because this method mainly depends on the data set when it is implemented, and can be separated from the support of the actual system when it is running, so it has a better advantage in the algorithm testing speed.

Offline experimental methods can be divided into prediction scoring accuracy and TopN recommendation. Based on the actual situation of teaching platform, compared with prediction scoring accuracy, TopN recommendation is more in line with the actual application needs, such as predicting whether students will watch a course or not, which is more important than predicting how much they will score after watching the course. TopN recommendation mainly predicts the accuracy of recommendation system through two metrics: precision and recall. The calculation formula is as follows:

(1) Accuracy

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

(2) Recall rate

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

In the above formula, R(u) is the recommendation list made to the student according to the learning behavior of the student in the training set, and T(u) is the behavior list of the student in the test set.

Here, we simply record the precision as P and recall as R. sometimes there are contradictions between these two indicators. For example, in extreme cases, if we only search out one result and it is accurate, then the precision is 100%, but recall is very low; If we return all the results, for example, recall is 100%, but precision will be very low. In this way, they need to be considered comprehensively. The most common method is F-measure (also known as F-score).F-measure is the weighted harmonic mean of precision and recall:

$$F = \frac{(\alpha^2 + 1)P * R}{\alpha^2 (P + R)}$$

When parameter $\alpha = 1$, which is the most common F1:

$$F1 = \frac{2P * R}{P + R}$$

It can be seen that F1 synthesizes the results of P and R. when F1 is higher, it can show that the test method is more effective.

The experimental data were collected from the online learning platform of our university. Firstly, the knowledge points in the online courses for the second semester of 2019-2020 school year (February-June, 2020) were prepared into a knowledge graph. Then, the learning and evaluation data of the 100 students that often utilized the platform were collected. These students are defined as those with more than 100 records of video viewing. The collected data were split into a training set and a test set. Then, a prediction experiment was carried out offline. The courses were recommended through collaborative filtering. To reflect recommendation accuracy, the recommendation results were compared with the data of the first semester of 2020-2021 school year (September 2020-January 2021).

After comprehensive adjustment, the weight values of each parameter of synergy index are as follows:

$$w_{collect} = 1$$
, $w_{like} = 0.9$, $w_{view} = 0.6$, $w_{class} = 1$, $w_{subject} = 0.9$,
 $w_{evaluate} = 0.7$, $w_{kp} = 0.12$, $w_{collaborative} = 0.18$

This paper analyzes the learning behavior data of the 100 students through the verification of the offline experimental method, and obtains the experimental data as shown in Table 3. It compares the performance of the independent collaborative filtering recommendation algorithm based on user similarity, the independent recommendation algorithm based on content and the improved hybrid recommendation algorithm in terms of accuracy and recall, It shows that the hybrid recommendation algorithm can achieve more effective recommendation. The data in the second and third columns of the table show the accuracy and recall rate of the user similarity algorithm using Euclidean distance, Pearson correlation coefficient and Spearman correlation coefficient under the condition of using UserCF and CBR recommendation algorithm separately. The experimental data show that the Spearman rank correlation coefficient is used to measure the user similarity in both User CF and CBR algorithms, the values of P and R are higher, and the fourth column data is the accuracy and recall rate obtained by using the hybrid recommendation algorithm designed in this paper. The data shows that the hybrid recommendation technology can recommend more effectively. The fifth column data is used to calculate the F1 value of hybrid recommendation technology. The experimental data show that the F1 value of "user similarity algorithm with Spearman correlation coefficient + hybrid recommendation technology" is 40.3, which is higher than that of the other two algorithms, so it can prove that the recommendation method is more effective.

Algorithm Similarity measure	User-C F	CBR	Hybrid recommendati on algorithm	F1 (based Hybrid recommendati on algorithm)
Euclidean distance	75.1/21. 3	70.2/11. 6	78.6/22.3	34.7
Pearson correlatio n coefficien t	79.3/23. 9	75.2/18. 5	83.2/24.2	37.5
Spearman 's rank correlatio n coefficien t	86.6/25. 8	78.8/20. 4	90.2/26.8	41.3

VII. CONCLUSIONS

Based on the data on student behaviors of an online teaching platform, this paper designs a personalized recommendation system for learning resources based on the collaborative filtering algorithm coupled with the knowledge point representation technology called knowledge graph. The high recommendation accuracy of our system was fully proved through experiment.

However, the authors have not deeply explored the influence of user portraits on system performance, due to the otherwise huge computing load of data preprocessing. But it is difficult to collect data and build the model, owing to the diversity of users. In future research, the authors will try to further improve the accuracy and recall of our algorithm.

ACKNOWLEDGMENT

This research was supported by 2020 Zhejiang Provincial Department of education general research project "Research on the talent training mechanism of social recruitment under the background of Higher Vocational enrollment expansion -- taking e-commerce major as an example" (No. Y202044895) and Zhejiang Vocational College of Commerce key teaching reform project "exploration of three education reform of e-commerce major under the background of double high plan" in 2020.

References

- [1] Tian Feng, Li Xin, et al. Learner model construction of personalized recommendation system for learning resources [J]. Education and teaching forum, 2020,3 (10): 304-305.
- [2] X. K. Zhao, S. R. Long, "Personalized recommendation of educational resources based on user learning path analysis", University Education, no. 10, pp. 107-110, 2019.
- [3] Y. J., Xu, J. Guo, "Recommendation of Personalized Learning Resources on K12 Learning Platform",

Table 3. Comparison table of similarity accuracy/ recall (unit: %)

Computer Systems & Applications, vol. 29, no. 7, pp. 217-221, 2020.

- [4] Y. G. Jia, "Design and implementation of an e-commerce recommendation system based on collaborative filtering", Digital communication World, no. 3, pp. 247, 2018.
- [5] C. Yang, L.H. Miao, B. Jiang, D.S. Li, D. Cao, "Gated and attentive neural collaborative filtering for user generated list recommendation", Knowledge-Based Systems, vol. 187, pp. 104839, 2020.
- [6] A. Mongia, N. Jhamb, E. Chouzenoux, A. Majumdar, "Deep latent factor model for collaborative filtering", Signal Processing, vol. 169, pp. 107366, 2020.
- [7] R. R. Wang, X. Ma, C. Jiang, Y. Ye, Y. Zhang, "Heterogeneous information network-based music recommendation system in mobile networks", Computer Communications, vol. 150, pp. 429-437, 2020.
- [8] X. Y. Huang, B. Liang, W. Li, "Online collaborative filtering with local and global consistency", Information Sciences, vol. 506, pp. 366-382, 2020.
- [9] Y. Zhao, "Design of Cyberspace Security Talents Training System Based on Knowledge Graph", International Journal of Digital Crime and Forensics, vol. 12, No. 4, pp. 44-53, 2020.
- [10] Z. Z. Wang, "iOS AutoTrack Solution", China Machine Press, August, 2020.

- [11] Z. Z. Wang, "Android AutoTrack Solution", China Machine Press, August 2020.
- [12] G. Z. Yang, "Comprehensive Treatment Service Platform of Research and Implementation Based on Massive Network Data (In Financial Field)", Jiaoyu Jiaoxue Luntan, no. 37, pp. 264-266, 2017.
- [13] J. Li, L. N. Dai, "Multiple Key Information Extraction Simulation Based on Integer Linear Simulation", Computer Simulation, no. 10, pp. 365-368, 383, 2020.
- [14] S. Harwood, "Introducing the VIPLAN Methodology (with VSM) for Handling Messy Situations – Nine Lessons", Systemic Practice and Action Research, 2020.
- [15] H. G. Wang, W. N. Fu, "Personalized Learning Resource Recommendation Method Based on Dynamic Collaborative Filtering", Mobile Networks and Applications, 2020.
- [16] Q. Liu, "Deep representation and intelligent application of educational resources", Artificial Intelligence, no. 3, pp. 44-54, 2019.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US