Online Course Recommendation Using Deep Convolutional Neural Network with Negative Sequence Mining

Ming Gao, Yonghan Luo, and Xiaonan Hu

1Computer School, Beijing Information Science & Technology University, Beijing 100101, China
2Department of Economics and Related Studies, University of York, York YO10 5DD, UK

Correspondence should be addressed to Ming Gao; gm@bistu.edu.cn

Received 11 May 2022; Revised 6 July 2022; Accepted 13 July 2022; Published 4 August 2022

Copyright © 2022 Ming Gao et al. 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.

Massive Open Online Course (MOOC) has been criticized for low completion rates, and one of the major reasons is that it fails to offer personalized course recommendations for different users with different demands. To solve this problem, this paper proposes a personalized course recommendation model based on convolutional neural network combined with negative sequence pattern mining. The model first models the course-learning sequence as a negative sequence pattern according to the user’s course registration, degree of completion, and final grades, in which, the negative term means that students should not choose and misoperate the principle of courses. Then, it employs a convolutional neural network structure to extract the internal features of negative sequence patterns for representation learning. Finally, through the convolutional sequence-embedding model, each user is recommended with a course list that includes the user’s maximized needs in recent temporal terms and the courses that are easy to be misselected. Experiment results show that the recommended model achieves higher recommendation performance with lower course dropout rate compared to baselines, which provides a new insight for both online and offline course recommendation.

1. Introduction

With the rapid development of online course technology and Internet technology, Massive Open Online Course (MOOC) has grown rapidly in recent years, attracting millions of online users [1–3]. For example, Coursera already has more than 4,300 courses and more than 200 university partners from 27 countries with over 50 million learners (as of 2022). According to the survey, MOOCs have educational benefits and 72% believed that MOOCs have professional benefits, which confirms the significance of online learning, especially during the epidemic of COVID-19 [5].

The issue of accurate personalized course recommendation in MOOC is the focus of both academia and industry [6, 7]. The roadmap of course recommendation is mainly divided into content-based (CB) recommendation [8, 9], collaborative filtering-based (CFB) recommendation [10–12], hybrid recommendation [13–15], and sequence-based (SB) recommendation [16–18]. Content-based recommendation helps to solve the problem of cold start, but it is greatly affected by data, often resulting in low recommendation accuracy [19, 20]. Collaborative filtering-based course recommendation focuses on mining the similarity of learners (i.e., users) and courses, with the basic assumption that similar users would take similar courses, but it can only make recommendations based on users with similar behaviours with target users, failing to solve the cold start problems, such as recommendation for new registered users [21, 22]. Hybrid recommendation is a combination approach by mixing content and collaborative filtering-based methods to help solve the problems of missing recommendation values and cold start [23–25]. Sequence-based approaches focus on mining learning behaviour sequences of users. SB methods are further mainly divided into two categories: the first one is to update data through sequences, and the other is to model learning behaviour sequences using time series technology [25].
Therefore, although the existing model has tackled the course recommendation problem existing in MOOC platform to a certain extent, MOOC still faces many new challenges. The high dropout rate is one of the most serious challenges facing MOOCs. According to statistics, the course completion rate of a MOOC platform is often less than 5% [1]. For users, when a large number of learning resources and activities are presented on the Internet at the same time, learners are inevitably confused by the overload of information resources, and it is difficult to quickly find learning resources suitable for them. Therefore, how to reduce the dropout rate and how to achieve personalized recommendation to users are the main research issues in the field of online learning recommendation [26].

In order to improve the personalized recommendation performance of MOOC learning platform, this paper proposes a personalized online course recommendation model based on convolutional neural network combined with negative sequence pattern mining by combining the preference of learning users, the correlation between course, and selection order of courses. The model first represents the course sequence as a negative sequence pattern according to the user’s course selection sequence, degree of completion, and final grades. Then, feature extraction of negative sequence pattern is carried out by convolutional neural network. Finally, a course list is recommended to each user through the convolutional sequence embedding model, which includes the user’s maximized needs in recent temporal terms and the courses that are easy to be misselected.

The rest of this paper is organized as follows: In Section 2, we make a brief review on course recommendation. Then, the detail of the proposed method is presented in Section 3. We performed experiments on real-world data and reported the results in Section 4. Section 5 gives the conclusion.

2. Related Work

With the rapid development of MOOC platforms, online learning resources are also sharply increasing. Due to the differences in cognitive ability and knowledge structure of learners, they cannot quickly identify and select the learning resources they are interested as well as demanded [27]. Therefore, it is urgent to make the intelligent model study how to recommend useful and interesting learning resources for learners efficiently and accurately. Early course recommendation models are mainly transformation of generalized linear models (including logistic regression and linear support vector machines) to predict learning behaviour of users [19]. For instance, Balakrishnan and Coetzee [28] proposed a hybrid model, which combined hidden Markov model (HMM) and logistic regression to predict the student retention rate of a single course. Pang et al. [29] proposed a multi-layer self-adaptive recommendation method to recommend courses on MOOC by combining the effectiveness and efficiency of a collaborative filtering recommendation method. Their method transforms the learner’s vectors into the same length dimension and disperses them into clusters containing similar learners with more common lessons. At the same time, the proposed model reduces the time overhead of online, offline, and update calculations in CF recommendation process. Later, Pang et al. [30] proposed an adaptive MOOC recommendation (ARM) solution to the problem of high dropout rate due to low satisfaction and loneliness on MOOC platforms. Their proposed method combines collaborative filtering with time series to improve the recommendation accuracy.

Deep learning methods have also been used to predict learning behaviours of online learners. For example, Fei and Yeung [31] combined sequence markers with an RNN-based model to predict the dropout probability of students. Wang et al. [32] proposed a hybrid deep neural network dropout prediction model by combining both CNN and RNN. In order to improve the learning efficiency and enthusiasm of learners, Zhang et al. [33] proposed a high-precision oriented resource recommendation model based on deep belief network (DBN) in the MOOC environment. In this method, the learner characteristics and course content attributes are deeply mined, and the user-course feature vector is constructed based on the learner behaviour characteristics as the input of the deep model. Later, Zhang et al. [34] proposed a course recommendation model, namely, MCRS, based on distributed computing framework to solve the problem that traditional recommendation system could not be directly and effectively applied to MOOC platform in a closed education environment with relatively stable number of courses and users. The basic algorithm of MCRS is an improved distributed association rule mining algorithm based on Apriori algorithm. It is also useful to mine course rules hidden in course registration data. Cristea et al. [35] proposed a light-weighted method that can predict users’ dropout situation before they start learning only according to their registration date. In addition to the prediction model itself, Nagrecha et al. [36] focused on the interpretability of existing dropout prediction methods. Dalipi et al. [37] reviewed dropout prediction techniques and made some insightful recommendations for this task. Qiu et al. [38] studied the relationship between student engagement as well as certificate rate and proposed a potential dynamic factor graph (LadFG) to model and predict learning behaviour in MOOCs. However, the current research has not considered the information gain that learners’ negative behaviour of course evaluation can bring to the recommendation model. In other words, the utility of introducing users’ negative behaviour to the course recommendation model in online learning platform needs further research.

3. Materials and Methods

This section describes data materials and the proposed recommendation model. We first introduce the acquisition of MOOC data and the representation of negative sequence. Then, we introduce the course recommendation model based on convolution sequence embedding. Figure 1 is the overall architecture diagram of the personalized online course recommendation model based on convolutional neural network combined with negative sequence pattern mining. The model is mainly divided into three modules, among which data preprocessing includes data acquisition,
Table 1: Symbol and denotations.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>$U = {u_1, u_2, \cdots, u_m }$ is the set containing $m$ users</td>
</tr>
<tr>
<td>$I$</td>
<td>$I = {i_1, i_2, \cdots, i_n }$ is the set containing $n$ courses</td>
</tr>
<tr>
<td>$\neg i$</td>
<td>Negative example course</td>
</tr>
<tr>
<td>$S^u$</td>
<td>$S^u = {i_{1u}, \neg i_{1u}, \cdots, \neg i_{nu} }$ represents the course selection sequence of the user</td>
</tr>
</tbody>
</table>

collation, and negative sequence construction. The convolutional neural network layer consists of embedding negative sequence patterns, convolution, and maximum pooling. The fully connected layer mainly connects user features with convolution sequences.

3.1. Dataset. In order to verify the effectiveness of the proposed method, this paper uses a web crawler to randomly crawl the usage information of 500 learner users who have studied no less than 20 courses in the internationally renowned course evaluation platform, namely, “Coursetalk.” The collected content includes the basic information of each user, course selection list, completion status, evaluation result, and final grades. Based on users’ overall rating and completion status of the course, this paper divides the course into positive cases and negative cases (i.e., truancy/dropout course). For example, if users’ completion of the course is less than 50%, this paper considers the course as a negative case. Considering that the range of users’ grading of courses is [0,10], the graded courses less than 5 are also defined as negative examples, and those greater than 5 are defined as positive examples. Then, according to the class time of each user, the user’s courses are arranged in sequence to form the class sequence of the user. The user’s course sequence is then embedded by deep neural network.

3.2. Preprocess of Data. Table 1 shows the main symbol and denotations of the proposed method. All users are represented as $U = \{u_1, u_2, \cdots, u_m \}$, where $U$ contains $m$ users, and each user represents a sequence. All courses are represented as $I = \{i_1, i_2, \cdots, i_n \}$, where $I$ contains $n$ courses. For a user $u$ who is associated with some courses in $I$, this paper uses the sequence $S^u = \{i_{1u}, \neg i_{1u}, \cdots, \neg i_{nu} \}$ to express such association, where $i_{1u}$ denotes the course which the user completion rate is greater than 50% and the score is higher than 5, namely, positive example course. Conversely, $\neg i_{nu}$ denotes negative example course, and $i_{1u}$ or $\neg i_{nu} \in I$. $t$ represents the order in which the user took lessons, in instead of the timestamp. For all user sequences $S^u$, the goal of this paper is to recommend to each user a list of courses that includes the user’s most recent needs and the ones that users are prone to misselection, by taking into account their general preferences and course selection ordering patterns.

The convolutional sequence embedding model uses the convolutional neural network to learn the characteristics of the user’s behaviour sequence and uses the latent factor model (LFM) to learn the user’s own characteristics of preferences. In order to train the network, for each user $u$, the model extracts every $L$ consecutive items (i.e., courses) from the user’s sequence $S^u$ as input and their next $T$ item as the target, as shown in Figure 1(a). The network establishes this by sliding a window with size $L + T$ on the user’s course selection sequence. Each window generates a training instance for $u$, represented by a triplet $(u, L, T)$, where $L$ represents the first $L$ items in each sliding time window of the user and $T$ represents the last $T$ items after the same time window.

3.3. Convolution Sequence Embedding-Based Course Recommendation. The proposed convolution sequence
embedding-based course recommendation model is mainly composed of three parts, namely, sequence embedding layer, convolution layer, and fully connected layer. Among them, the sequence embedding layer represents the user’s course selection activity record as a sequence and carries out vectorization; the convolution layer convolves the embedded sequence according to the sliding time window; the fully connected layer connects the user’s features with the convolution sequence so as to recommend the most suitable course sequence for users.

3.3.1. Sequence Embedding Layer. The convolutional sequence model captures sequence features in potential space by embedding the first L term into the neural network. For the embedding factor of item \( i \), \( q_i \in \mathbb{R}^d \), where \( d \) is the number of hidden dimensions, and similarly, the representation of users is denoted as \( P_u \in \mathbb{R}^d \). The embedding search operation retrieves the embeddings of the previous L terms as the embedding base and hews them together. For user \( U \) with time step \( t \), the result matrix is \( E^{(u,t)} \in \mathbb{R}^{L \times d} \), which is represented by

\[
E^{(u,t)} = \begin{bmatrix} Q_{u,h}^t & \cdots & Q_{u,L}^t \end{bmatrix}, \tag{1}
\]

3.3.2. Convolution Layer. The user behaviour sequence is obtained through the sequence embedding layer, and the convolution layer is used to process the sequence with length \( L \) after partition. If there are \( n \) filter layers, then \( F_k \in \mathbb{R}^{d \times 1} \), \( 1 \leq k \leq n \), \( h \in \{ 1, \cdots, L \} \) is the height of the convolution kernel. For example, assuming \( L = 4 \), then \( h = \{ 1, 2, 3, 4 \} \), and assuming that each \( h \) corresponds to two convolution layers, then \( n = 8 \). \( F_k \) slides from \( E \) from top to bottom. For the term \( i \), its sliding range is \( 1 \leq i \leq L - h + 1 \). The convolution process is formulated as

\[
C_i^k = \phi_i(E_{i+h-1} \odot F_k), \tag{2}
\]

where \( \odot \) denotes the convolution operation and \( \phi_i(\cdot) \) is the activation function. Equation (2) represents the inner product of the submatrix of \( F_k \) from the row \( i \) to row \( i + h - 1 \) of \( E \), i.e., \( E_{i+h-1} \). Therefore, the final result of \( F_k \) filter convolution is formulated as

\[
C_k = \left[ C_1^k, C_2^k, \cdots, C_{L-h+1}^k \right]. \tag{3}
\]

The max pooling operation is then applied to \( C_k \) to extract the maximum from all the values produced by this particular filter. Generally, the maximum captures the most important features extracted by the filter. Therefore, for \( n \) filters in this layer, the output value \( o \in \mathbb{R}^n \) is calculated by

\[
o = \{ \max (c^1), \max (c^2), \cdots, \max (c^n) \}. \tag{4}
\]

3.3.3. Fully Connected Layer. In order to obtain the general preferences of users, the basic characteristics of users are vectorized and expressed as \( P_u \) as we mentioned in the previous section. By concatenating \( P_u \) and \( o \) together, the fully connected layer projects them to the output layer with \( |I| \) nodes, which is formulated as

\[
y^{(u,t)} = w^T z + b', \tag{5}
\]

where \( w^T \in \mathbb{R}^{|I| \times 2d} \) and \( b' \in \mathbb{R}^{|I|} \) are learnable weight transformation and bias matrix, respectively.

3.3.4. Loss Function. To train the entire network, we transfer the \( y^{(u,t)} \) from output layer into recommendation probability as

\[
p(S_t^u | S_{t-2}^u, S_{t-3}^u, \cdots, S_{t-L}^u) = \sigma\left(y^{(u,t)}_{S_t^u}\right), \tag{6}
\]

where \( \sigma(x) = 1/(1 + e^{-x}) \) is the sigmoid function. \( C^u \) represents the set of time steps that the model predicts for user \( u \). The possibilities of all sequences in the dataset can be described by

\[
p(s | \theta) = \prod_u \sum_{s \in C^u} \sigma\left(y^{(u,t)}_{S_t^u}\right) \prod_{j \neq S_t^u} \left(1 - y^{(u,t)}_{S_t^u}\right), \tag{7}
\]

and thereby the loss function of the model is formulated as

\[
\mathcal{L} = \sum_u \sum_{s \in C^u} - \log \left(\sigma\left(y^{(u,t)}_{S_t^u}\right)\right) \prod_{j \neq S_t^u} \left(1 - y^{(u,t)}_{S_t^u}\right). \tag{8}
\]

4. Results

4.1. Evaluation Metrics. In this paper, we evaluate the proposed model comprehensively by using Precision@N, Recall@N, and mean average precision (MAP). During the test, the model predicts a course sequence which contains \( N \) courses for one user, denoted as \( R_{1,N} \), and we compare \( R_{1,N} \) with the last 20% course sequence which the user actually took and were masked in the training process, denoted as \( R \). Under these conditions, Precision@N and Recall@N for measuring the performance of recommendation are, respectively, calculated as

\[
\text{Precision@N}_{\text{rec}} = \frac{|R \cap R_{1,N}|}{N}, \tag{9}
\]

\[
\text{Recall@N}_{\text{rec}} = \frac{|R \cap R_{1,N}|}{|R|}. \tag{10}
\]

In order to distinguish between dropout strategy in deep learning and real incomplete courses, we use the term “truancy” to distinguish them. Suppose that the course sequence of truancy is \( M \) and the actual truancy course sequence is \( W \), then Precision@W and Recall@W for measuring the
truancy behaviour can be calculated as

\[
\text{Precision}_{\text{truancy}} = \frac{|W \cap M|}{|W|}, \tag{11}
\]

\[
\text{Recall}_{\text{truancy}} = \frac{|W \cap M|}{|M|}. \tag{12}
\]

Then, the mean average precision is calculated by

\[
\text{MAP} = \frac{\sum_{t=1}^{T} AP_t}{T}, \tag{13}
\]

where \( T \) denotes number of users in the test set. For one user, \( AP \) denotes the average precision which is calculated by

\[
AP = \frac{\sum_{N=1}^{R} \text{Precision}_{\text{truancy}} \times \text{rel}(N)}{|R|}, \tag{14}
\]

where \( \text{rel}(N) \) is the ratio of \( R_{1,N} \) and \( R \).

4.2. Baselines. In this paper, two representative course recommendation methods, namely, hybrid [39] and ensemble [40], were adopted to verify the effectiveness of the proposed method. The two models are briefly introduced below.

(i) **Hybrid Method** [39]. This method uses the basic information of courses and users as well as the actions taken by users to generate personalized recommendation of courses. The candidate courses were obtained by vectorization, weighting, and multirelational graph construction of the data collected from the website, and then, the courses were ranked according to their relevance to the target users by 3A ranking algorithm.

(ii) **Ensemble Method** [40]. This method can find patterns in user rating data and handle complex objects well. CBF attempts to recommend projects based on similarity in content, hence the project-to-project association approach. Project description and user positioning play an important role in CBF.

4.3. Results and Performance Evaluation. In this experiment, Equation (8) is used as the loss function for training the model, and the Adam optimizer was utilized [41]. As mentioned above, there are 500 online learning users in our dataset. In this paper, 100 users with truancy records (negative cases) are randomly selected as the test dataset, and the remaining 400 users are used for training model parameters. The model performance was evaluated by comparing the performance calculated by the aforementioned metrics. The main experimental results are shown in Table 2. As we observed from Table 1, when \( N \) is set to 5, the overall

<table>
<thead>
<tr>
<th>( N )</th>
<th>( P@N )</th>
<th>( P@W )</th>
<th>( R@N )</th>
<th>( R@W )</th>
<th>( P@N )</th>
<th>( P@W )</th>
<th>( R@N )</th>
<th>( R@W )</th>
<th>( P@N )</th>
<th>( P@W )</th>
<th>( R@N )</th>
<th>( R@W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.27</td>
<td>0.14</td>
<td>—</td>
<td>—</td>
<td>0.20</td>
<td>0.10</td>
<td>—</td>
<td>—</td>
<td>0.14</td>
<td>0.10</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>0.39</td>
<td>0.21</td>
<td>0.45</td>
<td>0.52</td>
<td>0.23</td>
<td>0.11</td>
<td>0.34</td>
<td>0.44</td>
<td>0.26</td>
<td>0.17</td>
<td>0.39</td>
<td>0.41</td>
</tr>
<tr>
<td>10</td>
<td>0.17</td>
<td>0.09</td>
<td>0.67</td>
<td>0.66</td>
<td>0.15</td>
<td>0.04</td>
<td>0.53</td>
<td>0.54</td>
<td>0.10</td>
<td>0.05</td>
<td>0.53</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Table 2: Recommendation performance among the proposed method and baselines.
highest precision rate of predicting on course recommendation is 0.391, and the corresponding recall rate is 0.210. Because the result is too incidental when the \( W = 1 \) for predicting course truancy, we only report the experiment results when \( N \) is 5 or 10. The precision and recall rates of the proposed method are much higher than the predicted result of the baseline models. Therefore, the effectiveness of the method proposed in this paper is significant, especially in predicting users’ truancy. Compared with hybrid method and ensemble method, the method presented in this paper improved by 35% and 92%, respectively, in Precision@1, indicating the proposed method has a significant effect in improving the order of recommended courses at first few items.

4.4. Parameter Study. In this paper, we further study the influence of hyperparameters. When testing the influence of \( d \) on the model, other parameters were freeze, and \( L \) was changed accordingly to obtain Figure 2(a). Similarly, the influence of \( d \) on the model is verified and shown in Figure 2(b). Figure 2(a) shows the change trend of MAP values of the model when \( T \) is 1, 2, and 3, respectively, with the increasing number of first \( L \) terms for the recommendation model. Figure 2(b) shows the change trend of MAP values corresponding to the model when \( T \) is 1, 2, and 3, respectively, with the increasing of potential dimension \( d \). As can be observed from Figure 2(a), the performance of the model first keeps rising with the increase of \( L \) and then tends to be flat. Among them, when \( L = 7 \) and \( T = 2 \), the performance of the model can reach the optimal. As we also observed from Figure 2(b), the performance of the model does not improve continuously with the increase of \( d \). When \( d = 50 \) and \( T = 2 \), the performance of the model derives the optimal performance.

In summary, by fixing other parameters used in this model and changing specific hyperparameter, we finally found that \( L = 7 \), \( T = 2 \), and \( d = 50 \) are the optimal configuration combinations.

5. Conclusion

In view of the shortcomings of high truancy rate and poor personalized recommendation effect in traditional online learning platforms, this paper combines convolutional neural network with negative sequence pattern mining into online course recommendation and proposes a personalized online course recommendation model based on convolutional neural network combined with negative sequence pattern mining. The model can not only recommend a list of courses with maximum demand in the near future but also predict which courses are most likely to be misselected by users. Experimental results show that the recommendation model proposed in this paper has a certain degree of course recommendation accuracy, and the method has a significant performance in predicting misselected courses, which provides a new thought and insight for online course recommendation.

Data Availability

In order to prevent the data from being abused, we open our data based on reasonable requests. Please contact the corresponding author with a formal application form to access the data.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors would like to thank Ms. Jun Yang for giving insightful suggestions for this work. This work was supported in part by the Professional Site Construction Project in Electronic Information (Computer Science) of Beijing Information Science and Technology University under Grant 5112211038.

References


