Research Article

Online Teaching Course Recommendation Based on Autoencoder

Dandan Shen¹ and Zheng Jiang²

¹School of Computer Science and Intelligence Education, Lingnan Normal University, Zhanjiang 524048, Guangdong, China
²Postgraduate School, Chengdu Sport University, Chengdu 610041, Sichuan, China

Correspondence should be addressed to Dandan Shen; shendd@lingnan.edu.cn

Received 9 June 2022; Accepted 5 July 2022; Published 31 August 2022

Academic Editor: Wenlong Hang

Copyright © 2022 Dandan Shen and Zheng Jiang. 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.

When using traditional recommendation algorithms to solve the problems of course recommendation, such as data sparseness and cold start, the performance of recommendation cannot be significantly improved. In order to solve its limitations in capturing learners’ preferences and the characteristics of courses, this paper first clarifies the research foundation of course recommendation based on autoencoder and analyzes the description of course relevance and recommendation methods. According to the timing characteristics of online learning, an online course recommendation model based on autoencoder is proposed where the long-term and short-term memory (LSTM) network is used to improve the autoencoder, so that it can extract the temporal characteristics of data. Then, the Softmax function is used to recommend courses. The experimental results show that, compared with recommendation model of collaborative filtering algorithm and traditional autoencoder, the proposed method has higher recommendation accuracy.

1. Introduction

In recent years, massive online open courses (MOOC) have gradually entered people’s field of vision with brand-new educational forms. MOOC provides learners with flexible ways and channels to acquire knowledge beyond the limitations of time, space and place, so that while mobile Internet technology has developed, and effectively promotes the development of online education [1, 2]. At present, in addition to completing the prescribed professional courses, students can choose their own courses according to their own interests and plans for the future. However, some students cannot make effective use of the course resources because they do not master enough about the courses and blindly follow the crowd. Therefore, how to choose suitable learners from many shared educational resources to take online courses has become one of the main problems that online learning platforms need to solve at present.

The recommendation system can effectively solve the problem of “difficult choice” caused by complicated learning resources in MOOC platform, and learners can conveniently obtain relevant course information, thus improving the personalized service level of the platform and greatly promoting the development of online learning. For online education, it can promote the development of online learning by using a recommendation system to provide personalized services for online learners, modeling learners’ preferences and providing recommended course services. Although personalized curriculum recommendation algorithms based on learners’ preferences have been proposed continuously, most of them have not yet been tapped [3, 4]. On the one hand, the problem of data sparseness in the recommendation system is particularly prominent, and further research is needed on other course recommendations, such as relying on the historical interaction data between learners and courses. On the other hand, the traditional recommendation model has limitations in capturing learners’ preferences and features, and the improvement of recommended performance is not significant.

Because of its powerful ability of nonlinear mapping, deep learning can effectively map high-dimensional data to low-dimensional space to extract high-level features. With the development of deep learning in recent years, personalized recommendation by deep learning has been widely
used [5, 6]. Among them, the autoencoder has strong recessive feature learning ability which can effectively manage the problem of data sparsity [7]. In addition, because users often consider the courses they have learned before, the sequence of user’s course selection has obvious temporal characteristics, and LSTM has powerful ability of temporal modeling, which can process time series data [8]. According to the above characteristics, LSTM is used to improve the autoencoder that is applied to the course recommendation field of online education, and the following courses are recommended according to the learning history of users.

2. Theoretical Basis

Recommendation system, as one of the technologies to solve information overload, is an effective defense against users over-selection, and can promote business of website and users’ decision making. The recommendation list generated by the recommendation system depends on user preferences, item characteristics, historical user-item interaction information, or other additional attribute information, such as time and space [9]. For online education, it can promote the development of online learning by using recommendation system to provide personalized services for online learners, modeling learners’ preferences, and providing recommended course services.

2.1. Description of Course Relevance. The emergence of deep learning technology has greatly innovated the recommendation system and brought more opportunities to improve the performance of recommendation, which can effectively capture the nonlinear interaction between learners and courses and mine learners’ preferences. However, another challenge is that the potential features of courses obtained from the training network model are mostly course sequences generated by random initialization, and the implicit preference model of learners cannot be constructed with fine granularity, which makes it difficult to distinguish the recommendation effect of different historic courses on learners’ preferences.

It is a common phenomenon that different courses have hierarchical relationships, and the relevance of courses is one of the factors that affect the recommendation effect. Pre-requisites mean that people must learn the concept of deeper knowledge before they are exposed to it, which can be regarded as very important to meet the learners’ curriculum experience, and it is also the premise to help learners choose their own courses according to the correct learning order. The pre-order and follow-up relationship of courses is not only for offline courses but also for online courses. However, the existing prerequisite relationships are mainly marked manually by teachers or domain experts [10], which cannot meet the increasing demand of online courses. On the other hand, online users may come from learners with different backgrounds, and different universities offer courses in different disciplines on MOOC, therefore, it is relatively difficult to make personalized learning for students from different backgrounds, and how to make full use of the attribute information of courses and capture the prerequisite relationship of online courses is challenging.

2.2. Course Recommendation Method. As shown in Figure 1, according to different recommendation methods, they can be divided into explicit feedback and implicit feedback.

The former makes use of learners’ scoring content of learned courses to predict the scores of other courses. The latter generates the corresponding course recommendation list according to the historical interaction information between learners and learned courses, such as the behavior data of learned courses. Compared with the problems such as false or random scoring in explicit feedback, recommendation based on implicit feedback is one of the commonly used recommendation algorithms, which can really mine the preferences of users. However, the model of matrix factorization can mine users’ features by simple combination, but cannot capture deeper feature vectors for complex data, which will greatly reduce the accuracy of recommendation and the convergence speed of the training process.

2.3. Long-Short Term Memory Network. LSTM is a variant of RNN network. The change lies in that it adds memory characteristics on the basis of RNN, which can endow neural networks with long-term memory ability and make the model have good applicability to long-term series.

Generally, it is only necessary to set up a four-layer structure to build the LSTM model, Input layer, LSTM layer, Fully connected layer, and Output layer.

The internal structure of LSTM is shown in Figure 2. The gate unit is expanded on the basis of RNN, and the complexity of the hidden layer is improved. The gate unit includes input stage, output stage, memory stage, and forget stage [11]:

1. Input stage: filter from input to decide whether to save the information.
2. Forget stage: determine whether the state information of the current neuron should be discarded.
3. Output stage: it determines whether the information state of the current neuron should be output.
4. Memory stage: the new input is filtered to determine which information can be preserved. Through the cooperation of different gate units in the hidden layer, the information is no longer updated indefinitely, but is filtered and removed in an orderly manner, which effectively solves gradient disappearance and realizes the preservation of long-time series data memory.

2.4. Autoencoder. Autoencoder is an artificial neural network that learns data coding by unsupervised learning. Its characteristic is that the learning goal is the same as the input data, and its general purpose is to reduce the dimension of data by training the neural network to ignore irrelevant information. Autoencoder consists of two parts: An encoder
that compresses and encodes the input data and a decoder that reconstructs the encoding into the input data. Figure 3 shows its basic structure.

The user’s rating data are input into the model, and the variables of the hidden layer are obtained after being processed by the encoder, and then the output data of the model are obtained after the variables of the hidden layer are reconstructed by the decoder. In the reconstruction process, the unknown item rating can be predicted.

\[
X_1, X_2, \ldots, X_n \text{ represents input data, } h_1, h_2, \ldots, h_n \text{ represents the data of the hidden layer, and the data of the output layer are expressed as } X'_1, X'_2, \ldots, X'_n. \text{ The encoder stage represents the part from the input layer to the hidden layer, where the input data } x \text{ of the dimension are encoded by the encoder function } f \text{ into a lower dimensional data } h \text{ of the hidden layer. For encoder mapping } \varphi \text{ and decoder mapping } \psi, \text{ there are}
\]

\[
\varphi: X \rightarrow F, \\
\psi: F \rightarrow X, \\
\varphi, \psi = \arg\min_{\varphi,\psi} \|x - \psi(\varphi(x))\|^2. \quad (1)
\]

In the encoding stage, \( x \in \mathbb{R}^d = X \) is mapped to hidden layer \( h \in \mathbb{R}^f = F, h \) is regarded as a hidden variable, and

\[
h = f(Wx + b). \quad (2)
\]

where \( f \) is the activation function, \( W \) is the weight matrix of input \( x \), \( b \) is the bias element. \( W \) and \( b \) are updated by backpropagation algorithm during training. In the decoding stage, \( h \) is mapped to the output, \( x' \) and \( x \) have the same structure:

\[
x' = f'(W'h + b') \quad (3)
\]

Train the self-encoder by minimizing the reconstruction error, as shown in the following equation:

\[
L(x, x') = x - x'^2 = x - f'(W'(f(Wx + b)) + b'))^2, \quad (4)
\]

where \( W \) is the weight matrix of input \( x \), \( b \) is the bias element.

3. Online Course Recommendation Model Based on Autoencoder

3.1. Acquisition of Data Set. Although the online data set of the university contains the brief contents of some related courses, the brief contents of some courses are too cumbersome, which is not conducive to the subsequent natural language processing to extract the relationship between courses. Therefore, part of the text information is obtained by the crawler tool on the website. The specific steps are as follows, as shown in Figure 4.

\[
A \xrightarrow{X_{t-1}} X_t \xrightarrow{X_{t+1}} A \\
A \xrightarrow{X_{h-1}} X_h \xrightarrow{X_{h+1}} A
\]

Figure 1: Different ways of recommending courses.

Figure 2: Internal structure diagram of LSTM.

Figure 3: Structure of autoencoder.

\[
A \xrightarrow{X_{t-1}} X_t \xrightarrow{X_{t+1}} A \\
A \xrightarrow{X_{h-1}} X_h \xrightarrow{X_{h+1}} A
\]
(1) Open the online website of the university, and learners can view the pages of the corresponding course introduction content, which are converted into HTML-encoded pages by browser tools, and the ways of data compilation in different positions of the pages are analyzed;
(2) Use the regular expression to obtain the text data of the corresponding course;
(3) Store the descriptions of these courses in the local database according to the uniform format, and then clean the data.

In this paper, a real data set of MOOC is adopted [12], which contains 458,454 records of course selection, including 82,535 users and 1,302 courses. Each record contains user id, course selection time, course id, course name, course category, and other attributes.

In order to better verify the follow-up experiments, this paper will do the following operations on the noise data:

(1) Simplify the original data into triples (user id, course selection time, course id);
(2) Among the 82,535 users, those with less than 10 elective courses are excluded, leaving 130,812 pieces of data and 8,268 users;
(3) Divide users into 8,268 groups, and sort in each group according to the ascending order of selection time;
(4) Merge the data in the group according to user id, course id time series, so as to obtain the final data set, totaling 8,268 pieces of data.

### 3.2. Time Series Feature Extraction Model

In order to model the time series of the user’s course selection data, LSTM is used to replace the feedforward neural network in the autoencoder, which is similar to RNN Encoder Decoder and ENDEC-AD. The proposed time series feature extraction method is shown in Figure 5.

The method proposed in this paper includes five parts: input layer, coding layer, LSTM layer, decoding layer, and output layer. First, the input data are preprocessed, including data enhancement and sequence division according to the window, then each sequence is sent to the encoder to obtain the feature vector, which is used as the input of LSTM network, and finally the output of LSTM is sent to the decoder, so as to get the reconstruction of the next window of the original data. The training is performed by minimizing the reconstruction error, and after the training is completed, the LSTM network output is extracted as the extracted time series feature.

1. Preprocess the original time series data, and input them into the input layer.

In the preprocessing process, data enhancement technology is adopted, and Gaussian noise is added to improve the accuracy of the model, as shown in the following equation:

$$X' = X + \gamma \cdot \sigma^2 \cdot \text{randn}(m),$$  \hspace{1cm} (5)

where $X'$ is the enhanced data, $X$ is the original data, $\gamma$ is the proportion of noise, $\sigma$ is the length of the original data, randn$(m)$ can produce $m$ data subject to the standard normal distribution. Then, the data with added noise are sequentially divided into windows with length $p$.

$$w_t = [x_{t-p+1}, x_{t-p+2}, \ldots, x_t].$$  \hspace{1cm} (6)

Among them, $x_t \in X'$, $w_t$ which indicates the data sequence whose end time is $t$.

2. Divide $k$ non-overlapping windows sequentially, and the encoded features $E_i$ are regarded as the input of LSTM network.

$$W_i = [w_{t-(k-1)p}, w_{t-(k-2)p}, \ldots, w_t],$$  \hspace{1cm} (7)

$$e'_i = \text{Encoder}(w'_i), \text{ for } i = 1, 2, \ldots, k,$$  \hspace{1cm} (8)

$$E_i = [e'_1, e'_2, \ldots, e'_k],$$  \hspace{1cm} (9)

where $W_i$ represents the window sequence whose end time is $t$, $e'_i$ represents the feature representation of the $i$-th window $w'_i$ in $W_i$, and $E_i$ represents the feature representation corresponding to window sequence $W_i$.

3. Train LSTM network, the first $k-1$ window in a sequence $E_i$ is used as input data to predict the next $k-1$ windows, as shown in the following equation:

$$[e'_2, e'_3, \ldots, e'_k] = \text{LSTM}([e'_1, e'_2, \ldots, e'_{k-1}]).$$  \hspace{1cm} (10)

Then, the calculation of LSTM can be expanded according to the following formulas:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f),$$  \hspace{1cm} (11)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i),$$  \hspace{1cm} (12)
\[ C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \]  
(13)

\[ C_t = f_t \ast C_{t-1} + i_t \ast \tilde{C}_t, \]  
(14)

\[ O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \]  
(15)

\[ h_t = O_t \ast \tanh(C_t), \]  
(16)

where \( W_f, W_i, W_c, W_o \) and \( b_f, b_i, b_c, b_o \) are weighted moments and bias vectors of forgetting stage, input stage, cell state, and output stage, respectively, in LSTM unit at time \( t \). \( \sigma \) is activation function, \( h_{t-1} \) indicates the state of the hidden layer at \( t-1 \) time.

Gradient clipping is applied to solve the problem of gradient explosion in LSTM, and the gradient value is kept at a reasonable level by setting the gradient exceeding the threshold as a fixed value, as shown in formula (17).

\[ \text{if} \| g \| > v \quad g = \frac{g v}{\| g \|}. \]  
(17)

Among them, \( v \) is the norm of the threshold, \( g \) is the parameter to be updated.

(4) Use a decoder to reconstruct the output of LSTM into a time series window.

\[ \hat{w}_{t-(k-i) \times p} = \text{Decoder} (\tilde{c}_t), \quad \text{for } i = 2, 3, \ldots, k. \]  
(18)

3.3. Course Recommendation Model. After extracting the time series characteristics of the data in the previous step, the output of LSTM is used as the input, and the final recommendation result of the model is obtained through the Softmax function, as shown in the following equation:

\[ y_u = \text{Softmax} (W_u \cdot h + b_u). \]  
(19)

The recommendation process of online courses is shown in Figure 6.

3.4. Model Training. This paper adopts the implicit feedback recommendation method of students’ behavior data, and studies the behavior data records of learners’ learning courses. It can be found that the main content is the relevant information of their learning courses. If a student likes one of the courses, then his frequency of learning this course will increase. Therefore, in this paper, it is considered that the interactive data of students enrolled in courses will show the data records of the courses they have studied as a positive example of training. Another problem is that these learners’ behavior data lack negative examples, which leads to the inability to obtain learners’ truer preferences from the registered course data. Therefore, we adopt the weighting method to distinguish learners’ learned courses from those that have not been learned, that is, all the courses that students have not learned are regarded as negative examples of training, and these negative examples are uniformly set to the same weight value. When learners’ frequency of learning courses increases, the corresponding weight value will also
increase, and the increasing way will show a proportional trend. According to this method, a confidence matrix $C$ of learners’ preferences can be proposed, in which the weights of elements in the matrix are set as follows:

$$c_{s,c} = \begin{cases} 1 + \alpha \log(1 + f_{s,c}/\epsilon), \\ 1, \end{cases}$$

where $\alpha$ and $\epsilon$ represent the confidence weight parameters. $f_{s,c}$ is the weight value of positive example calculated when the learner’s learning frequency is greater than 0, and the weight value of $c_{s,c}$ is set to 1 in other conditions.

After embedding the confidence matrix, the objective function of the model proposed is as follows:

$$\text{Loss} = \sum_{s=1}^{M} \sum_{c=1}^{N} \left\| C \Phi \left( X_{s,c} - \tilde{X}_{s,c} \right) \right\|_2^2.$$  \tag{21}

Because the regularization method is not only robust but also has a higher generalization effect, which can alleviate the noise data well. Therefore, by embedding the regular term, the above objective function can be rewritten as follows:

$$L = \text{Loss} + \lambda \left( \| W_a \|_2^2 + \| W_i \|_2^2 + \| W \|_2^2 \right).$$ \tag{22}

where $\lambda$ represents the regularization parameter, and $i \in \{1, 2, 3, 4\}$ in $W_i$ represents the corresponding weight matrix respectively. $W_a$ and $W_i$ are learning parameters.

4. Experiment and Discussion

4.1. Evaluation Indicators. In this paper, three evaluation indexes commonly used in the recommendation system, namely precision ($P$), recall ($R$), and $F1$, are used to measure the recommendation performance of the proposed method [13]. The equations are as follows:

$$P = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|},$$ \tag{23}

$$R = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|},$$ \tag{24}

$$F_1 = \frac{2 \times P \times R}{P + R}.$$ \tag{25}

where $R(u)$ represents the list of course recommendations made to users based on the data of course learning on the training data set, and $T(u)$ represents the list of course learning on the test data set.

4.2. Experimental Results. This paper takes $N \in \{1, 3, 10\}$ and sets up three groups of comparative experiments. In each group of comparative experiments, it is compared with a collaborative filtering algorithm (CF) and a recommendation algorithm based on traditional Autoencoder (AE). The results are shown in Figure 7:

In the same group of comparative experiments, the performance of the proposed method in $P$, $R$, $F1$, three indicators are better than collaborative filtering algorithm and recommendation algorithm based on traditional autoencoder. When $N = 1$, compared with CF algorithm, $P$ increases by 22.8%, $R$ increases by 26.2% and $F1$ increases by 24.6%. Compared with AE algorithm, $P$ increases 9.9%, $R$ increases 7.7%, and $F1$ increases 8.7%. While $N = 3$, compared with CF algorithm, $P$ increases by 15.5%, $R$ increases by 12.7%, and $F1$ increases by 14.1%. Compared with AE algorithm, $P$ is increased by 8.5%, $R$ increased by 7.4%, and $F1$ increased by 7.9%. In addition, when $N = 10$, compared with CF algorithm, $P$ increases by 13.3%, $R$ increases by 14.2%, and $F1$ increases by 13.8%. Compared with AE algorithm, $P$ increases by 3.1%, $R$ increases by 2.6%, and $F1$ increases by 2.9%. Experimental results show that the performance of a recommendation algorithm based on a deep learning method is better than that of a traditional algorithm, because it can extract deeper data features. Moreover, because LSTM is used to model the time series characteristics of data, the proposed method is better than the traditional autoencoder based recommendation algorithm.

In addition, among different contrast experimental groups, with the increase of $N$, the three performance evaluation indexes of the model become higher and higher, because with the increase of the number of recommended courses, the probability of including courses that users like is also increasing. To sum up, the experimental results on real data sets show that the autoencoder based on curriculum relevance can improve the overall evaluation index of course recommendation to a certain extent. Contrasted and other gauge suggestion calculations, the proposed strategy enjoys benefits in execution and can tackle the issues of unfortunate
versatility and data sparseness of recommendation algorithms. In addition, the designed model not only captures learners’ preferences but also accurately predicts learners’ comprehensive rating information in combination, which avoids recommendation bias caused by data sparseness, and further improving the adaptability of course recommendation results in online learning platforms.

5. Conclusion

Aiming at the problem of course recommendation in online education, this paper proposes an online course recommendation model based on improved autoencoder. First, the LSTM network is used to improve the autoencoder, so that the model can extract the temporal characteristics of data. Then, the extracted features of data are used to recommend courses for online education. Finally, experiments on real MOOC data sets show that the proposed method has higher accuracy compared with collaborative filtering algorithm and traditional autoencoder based recommendation algorithm, and it can accurately predict learners’ comprehensive rating information by combining the course relevance factors, thus avoiding the problem of recommendation bias caused by data sparseness, which further improve the adaptability of course recommendation results in online learning platform.

Data Availability

The data set can be accessed upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the 2019 Philosophy and Social Science Planning Project of Guangdong Province “Research on The Sharing Mode of Higher Education Information Resources Based on the Concept of Block Chain”; the project number is GD19CYJ19 and 2021 Annual Education Scientific Planning Project of Guangdong Province (higher education program) “Colleges and Universities Online Education Resources Construction and Application Research of Deep Learning Horizon”; the project number is 2021GXJK351.

References


