Doi: 10.30473/IDEJ.2022.63913.1104 **ISSN:** 2588-4476

Iranian Distance Education Journal

Vol. 3, No. 2, (New Series) Summer-Autumn 2021 (P 78-84), Payame Noor University

Original Article

An Implicit Feedback Recommendation System for Massive Open Online Courses

Parham Moradi [*1], Azadeh Faroughi[2]

1. Associate Professor ,Department Computer Engineering ,Faculty of Engineering, University of Kurdistan Sanandaj, Iran
2. Educator, Department Computer Engineering,Faculty of Engineering, University of Kurdistan Sanandaj, Iran

Received: 2021/06/22 **Accepted:** 2021/10/22

Abstract

Massive open online courses (MOOCs) have recently becoming a popular means of education. They generally give the students large-scale options. However, the diversity of MOOC courses available and their rapid updates make it more difficult for students to find fresh material relevant to them. A recommendation system (RS) connects the learner with the best learning resources to meet students' interests. The majority of recommender system research is based on the existence of explicit feedback, which is often impossible or inaccessible in MOOCs. As a result, in this paper, we model user positive and negative preferences using implicit feedback acquired passively by watching various types of students' behavior. This paper proposes a novel course recommendation, which employs Siamese Neural Networks (SNNs) to extract latent representations of students and courses using a loss function that favors observed over unobserved courses. The similarity of users and courses is then determined using a novel representation mechansim.

Keywords

MOOCs, Implicit Feedback, Recommendation System, Siamese Neural Network, Content Information

Introduction

Recently, massive open online courses, or MOOCs, are becoming increasingly popular as a form of alternative education. Many MOOC platforms have been built to provide low-cost access to many courses from the world's finest universities, such as Coursera, edX, and Udacity. Recommendation technology assists users in finding their favorite content from the massive amounts of data available, even if they have little knowledge or experience with the subject matter. Profiles include various characteristics of users and items, such as user (like age, gender, education) and item (like actor, genre, and category in the case of a Movie item) (Lops, Gemmis, & Semeraro, 2011). Hybrid combines elements of both collaborative filtering and content-based. Some RSs present related information in the form of a priority list, prioritizing items or services that are more relevant to the user's interests above resources that are less relevant to the user. Users are recommended commodities by well-known platforms such as "amazon.com" and "taobao.com" based on their browsing and purchasing history (Linden, Smith, & York, 2003). To suggest videos to their clients, "Netflix.com" and "Youtube.com" use recommendation tourniquets (Bennett & Lanning, 2007)(Davidson et al., 2010).

There are many MOOC recommenders proposed in the literature. Sparsity and cold start are two major issues of these methods. Sparsity means limited data available for the algorithm to provide accurate recommendations. On the other hand, there are only a few similar students for a student to be exploited in the recommendation process. Moreover, the cold-start students/courses refers to a situation where there are a few available ratings (or feedback) for a student or a course.

*Corresponding Author: p.moradi@uok.ac.ir

In this paper, we proposed an algorithm that recognizes users' latent information based on their implicit behaviors to overcome the sparsity issue. We look into student behavior and determine his interest based on the time spent watching course videos. A student may have enrolled in a course for various reasons, and it may not be his preferred course. To this end, in this paper, a Siamese Neural Network (Koch, Zemel, & Salakhutdinov, 2015) was used to compute the similarity between a student and a course. This method uses the triplet loss function, in which a student vector serves as an anchor to be compared by both of his liked and unliked courses . The proposed method assigns higher ranks to those of positive courses (liked ones) than negative ones. The proposed method employs the profiles of students and courses as an effective side information in its process to tackle the sparisty and cold-start issues. The variety and amount of recommendation outcomes are boosted as a result of this content data, which exposes users' potential choices more clearly.

Proposed Method

This section aims to present a novel method to tackle the Sparsity and cold-start issues of MOOC recommenders. The proposed method include several steps. First, users' implicit feedback is extracted based on a sequence indicating a user's history of click activities. The users' interests are defined based on the length of time they spend watching course videos. Then, a deep learning mechanism is used to extract the representation of students and courses. Finally, these representations are used to compute the similarity of courses and items, and thus, similar courses are recommended to students.

-Problem Definition

There are two main aspects to our course recommendation problem: user and item. Each item in XuetangX represents an open course, and each user shows an enrolled student. The set of users is denoted by U, the set of courses is denoted by C and |U| = n, |C| = m represents the number of users and courses, respectively. The set of all relations is A={(u,c)|u∈U,c∈C}, which shows user u's preference for course c. According to how much engagement there was, this research focuses on user interaction with a course. As a result, the a_UC matrix element is defined as:

$$a_{uc} = \begin{cases} 1, & if\ Duration(watching\ \ c\ by\ u) > threshold, \\ 0, & otherwise. \end{cases} \tag{1}$$

where, the duration of watching a video is determined according to the sequence of data records (load video, play video, pause a video, or stop it) in our dataset. For each user, the time spent for each course is calculated based on the total time spent watching the video at different times.

-Extract Embeddings

The proposed method aims to uncover previously unseen connections between user u and course c. As a result, the embedding of components is first extracted based on interactions, and then the interactions are recreated based on new representations in the prediction step. In the embedding step, users and items are transformed to vectorized representations, where they are mapped into a shared space S∈R^d. We proposed a deep-learning architecture to determine the embedding space. This model learns a nonlinear transformation that projects the instances into a target space where the distance between the positive and negative courses is smaller than the distance between the negative and positive courses. The proposed method uses the Siamese network to represent students and courses to fixed-size vectors. The Siamese framework has two identical feed-forward multilayer neural networks comprised of dense layers of rectified linear activation units (ReLUs). Fig.1 shows a Siamese network (Koch et al., 2015) used in the proposed method. According to this figure, the input to the Siamese framework for each training instance is a specific user as an anchor and two courses with ‘one and zero’ label which indicate whether or not the user interacts with these courses, respectively.

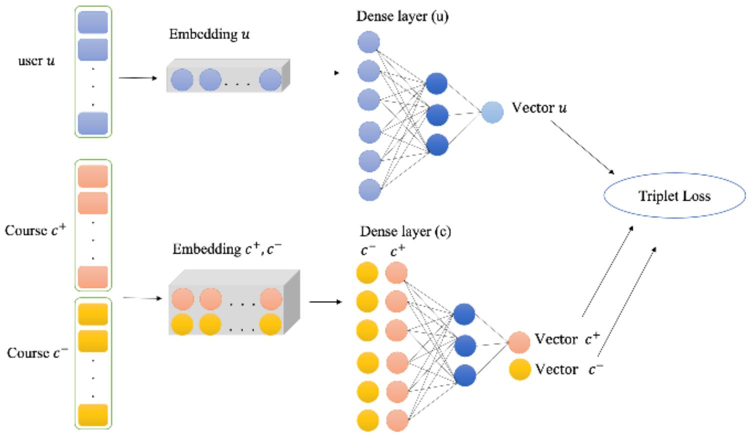

Figure 1. Embedding with Siamese Neural Network

According to this figure, c^+ and c^- are the positive course (interacted) and negative one (not interacted) by user u, respectively. Therefore, a training set of triplets can be expressed as:

$$T=\{(u,c^+,c^-) \mid a_{uc^+}=1, a_{uc^-}=0\} \quad (2)$$

The embedding layer encodes inputs as a dense vector (into d-dimensional vectors in the space S) through a single-layer MLP (i.e., fully connected neural network). A single layer MLP transforms the inputs into a dense vector:

$$e_u=\pi_u W_u, e_{c^+}=\pi_{c^+} W_c, e_{c^-}=\pi_{c^-} W_c \quad (3)$$

where $W_u \in R^{n\times d}$, $W_c \in R^{m\times d}$ are the transformation matrix, which invert the inputs of different dimensions into the same dimension vector in the shared potential space, d denotes the embedding size. In this equation, e_u indicates user embedding e_u, e_{c^+} and e_{c^-} denote positive course embedding and negative course embedding, respectively. Positive and negative course embeddings share the same embedding layer. If we don't have content information of the user and items, we use one-hot vectors of $\pi_u \in R^{n\times 1}, \pi_{c^+} \in R^{m\times 1}, \pi_{c^-} \in R^{m\times 1}$ as inputs. However, we can replace the one-hot encoding with an attribute vector when the student and course attributes are available. A triplet loss function guides the training of the embedding model. A triplet loss compares a reference input termed anchor (user u) to a matching input (positive course c^+) and a non-matching input (negative course c^-) to produce a loss function. The anchor's distance from the positive input is kept to a bare minimum, while the distance between the anchor and the negative input is increased. For this, we use dot product loss functions, where embeddings of courses are multiplied by a user embedding. The dot product loss originates from BPR-Opt (Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2012).

$$loss(\hat{u},c^+,c^-)=\ln(1-\sigma(\langle \hat{u},c^+\rangle-\langle \hat{u},c^-\rangle)) \quad (4)$$

where, σ is the sigmoid function and $\langle . \rangle$ implies the dot product computation. To optimize the model, stochastic gradient descent (SGD) is used. In each step of optimization, a batch of triplets is randomly selected from T. To make the optimization effective, during training hard examples of triplets (i.e. nearest negative course to user) is selected (Schroff, Kalenichenko, & Philbin, 2015).

-Prediction

After extracting the users and courses embeddings, we conduct the inner product to estimate the student's preference toward the target course:

$$\hat{y}_{uc} = e_u^T e_c \qquad (5)$$

where, $\hat{y}_{uc}$ denotes the probability that user u prefers course c.

Experiments

-Dataset

The dataset used in our study consists access and enrollment logs collected during September 2016 and October 2016 from XuetangX. The access logs include log times of sequence user's actions like 'load_video', 'play_video', 'pause_video', 'stop_video' and 'problem_get'. We take the data in September 2016 as the training set of our framework and data in the first week of October 2016 as the test set to validate the quality of course recommendation. We sort the activity of each user by their time stamp and define the interaction network based on duration of watching videos (according to Eq. (1)) with threshold 30 minutes. Table. 1 demonstrates the lists of basic information of dataset.

Table 1. Dataset Description

Type	Number	Content Information
Users	13884	Age, Gender, Education
Courses	245	Category
Interactions	38512	-

Evaluation Metrics

To evaluate our method, we use four accuracy metrics, including precision, recall, hit rank (HR) and average reciprocal hit rank (ARHR) which are widely used in top-N recommendation. For a user u, the top N recommended courses are expressed as $P_u = \{c_1, c_2, \ldots, c_N\}$ and the ground truth items are denoted as $G_u = \{c_1^{'}, c_2^{'}, \ldots, c_{|G_u|}^{'}\}$, where $|.|$ is the cardinality of set.

Results

We conduct experiments with two aspects of the proposed method: the first does not use content information of users and courses and only considers user interactions on courses, and we call it Siamese MOOC Recommender (SMR), and the second includes profile information in training the algorithm and is called Siamese MOOC Recommender with Context (SMRC).

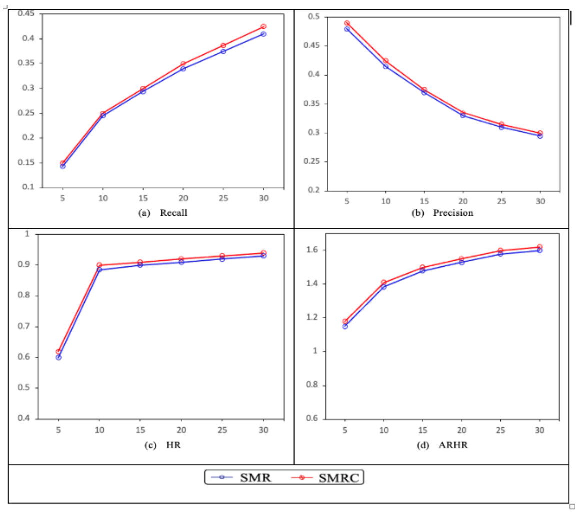

Figure 2. Top-N recommended courses (X-axis) vs. accuracy metrics (Y-axis)

Figure. 2 shows recall, precision, HR, and ARHR versus the number of top-N recommendation courses, N ∈{5, 10, 15, 20, 25, 30} for each algorithm. Each plot shows a particular metric. The results demonstrate that the embedding models with content information outperformed the other one which just consider the user behaviors. We compare our methods with four recommendation systems.

We set the number of recommender courses at N = 15 for all methods. Tables 2 demonstrate the performance of these methods on our dataset in terms of accuracy metrics. Numbers in boldface indicate the best accuracy in each column. Compared with other methods the proposed method can characterize variant user-course interactions well.

Table 2. Comparison with CF methods

	Recall	**Precision**	**HR**	**ARHR**
CoFactor	0.3245	0.3342	0.9134	1.5367
Metapath2vec	0.2534	0.2643	0.8234	1.3235
BiNE	0.1534	0.1834	0.6923	0.8543
SMR	0.3535	0.4824	0.9734	1.9256
SMRC	0.3823	0.4944	0.9522	1.9567

This paper looks into the MOOCs platform's course recommendation problem. We employed implicit feedback to overcome the sparsity problem, allowing the model to actively gather more interactive information from the system in order to deliver better course recommendations. We take into account each user's time spent on the course as an indicator of his or her level of interest in the course. Furthermore, we proposed using SNN to determine user and course similarity in a unique spatial representation. In addition, we employed user and course content information to address the issue of cold start, resulting in more realistic representations. We tested our methods against a variety of recommendation systems and found that the proposed method outperforms a number of standard algorithms. It would also be great to look into how other sorts of content, such as video and knowledge points, may be recommended in MOOCs to fulfill a range of individualized needs.

References

[1] Aher, S. B., & Lobo, L. (2013). Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data. Knowledge-Based Systems, 51, 1–14.

[2] Bennett, J., & Lanning, S. (2007). The Netflix Prize. KDD Cup and Workshop, 3–6.

[3] Breslow, L., Pritchard, D. E., DeBoer, J., Stump, G. S., Ho, A. D., & Seaton, D. T. (2013). Studying learning in the worldwide classroom research into edX's first MOOC. Research & Practice in Assessment, 8, 13–25.

[4] Dakhel, G. M., & Mahdavi, M. (2013). Providing an effective collaborative filtering algorithm based on distance measures and neighbors' voting. International Journal of Computer Information Systems and Industrial Management Applications, 5, 524–531.

[5] Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., others. (2010). The YouTube video recommendation system. Proceedings of the Fourth ACM Conference on Recommender Systems, 293–296.

[6] DeBoer, J., Stump, G. S., Seaton, D., Ho, A., Pritchard, D. E., & Breslow, L. (2013). Bringing student backgrounds online: MOOC user demographics, site usage, and online learning. Educational Data Mining.

[7] Dong, Y., Chawla, N. V, & Swami, A. (2017). metapath2vec: Scalable representation learning for heterogeneous networks. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 135–144.

[8] Gao, M., Chen, L., He, X., & Zhou, A. (2018). Bine: Bipartite network embedding. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 715–724.

[9] Gong, J., Wang, S., Wang, J., Feng, W., Peng, H., Tang, J., & Yu, P. S. (2020). Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 79–88.

[10] Guo, P. J., & Reinecke, K. (2014). Demographic differences in how students navigate through MOOCs. Proceedings of the First ACM Conference on Learning@ Scale Conference, 21–30.

[11] Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. ICML Deep Learning Workshop, 2. Lille.

[12] Liang, D., Altosaar, J., Charlin, L., & Blei, D. M. (2016). Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. Proceedings of the 10th ACM Conference on Recommender Systems, 59–66.

[13] Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76–80. https://doi.org/10.1109/MIC.2003.1167344

[14] Lops, P., Gemmis, M. de, & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. Recommender Systems Handbook, 73–105.

[15] Manouselis, N., Drachsler, H., Verbert, K., & Duval, E. (2012). Recommender systems for learning. Springer Science & Business Media.

[16] Pang, Y., Liao, C., Tan, W., Wu, Y., & Zhou, C. (2018). Recommendation for MOOC with learner neighbors and learning series. International Conference on Web Information Systems Engineering, 379–394. Springer.

[17] Piao, G., & Breslin, J. G. (2016). Analyzing MOOC entries of professionals on LinkedIn for user modeling and personalized MOOC recommendations. Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, 291–292.

[18] Qiu, J., Tang, J., Liu, T. X., Gong, J., Zhang, C., Zhang, Q., & Xue, Y. (2016). Modeling and predicting learning behavior in MOOCs. Proceedings of the Ninth ACM

International Conference on Web Search and Data Mining, 93–102.
[19] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence 452–461.
[20] Sandeen, C. (2013). Integrating MOOCs into traditional higher education: The emerging “MOOC 3.0” era. Change: The Magazine of Higher Learning, 45(6), 34–39.
[21] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 815–823.
[22] Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating “word of mouth.” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 210–217.
[23] Sidi, L., & Klein, H. (2020). Neural Network-Based Collaborative Filtering for Question Sequencing. ArXiv Preprint ArXiv:2004.12212.
[24] Toscher, A., & Jahrer, M. (2010). Collaborative filtering applied to educational data mining. KDD Cup.
[25] Wilkowski, J., Deutsch, A., & Russell, D. M. (2014). Student skill and goal achievement in the mapping with google MOOC. Proceedings of the First ACM Conference on Learning@ Scale Conference, 3–10.
[26] Zhang, H., Yang, H., Huang, T., & Zhan, G. (2017). DBNCF: Personalized courses recommendation system based on DBN in MOOC environment. 2017 International Symposium on Educational Technology (ISET), 106–108. IEEE.