

Representing and Predicting Student Navigational Pathways in Online College Courses

Renzhe Yu
School of Education
University of California, Irvine
Irvine, CA, 92697, USA
renzhey@uci.edu

Daokun Jiang
Dept. of Computer Science
University of California, Irvine
Irvine, CA 92697, USA
daokunj@uci.edu

Mark Warschauer
School of Education
University of California, Irvine
Irvine, CA 92697, USA
markw@uci.edu

ABSTRACT

Representation and prediction of student navigational pathways, typically based on neural network (NN) methods, have seen their potential of improving instruction and learning under insufficient human knowledge about learner behavior. However, they are prominently studied in MOOCs and less probed within more institutionalized higher education scenarios. This work extends such research to the context of online college courses. Comparing student navigational sequences through course pages to documents in natural language processing, we apply a skip-gram model to learn vector embedding of course pages, and visualize the learnt vectors to understand the extent to which students' learning pathways align with pre-designed course structure. We find that students who get different letter grades in the end exhibit different levels of adherence to designed sequence. Next, we fit the embedded sequences into a long short-term memory architecture and test its ability to predict next page that a student visits given her prior sequence. The highest accuracy reaches 50.8% and largely outperforms the frequency-based baseline of 41.3%. These results show that neural network methods have the potential to help instructors understand students' learning behaviors and facilitate automated instructional support.

Author Keywords

Online Course; Student Modeling; Representation Learning; Navigational Prediction; Skip-Gram; Recurrent Neural Networks; Long Short-Term Memory

INTRODUCTION

With the prevalence of online learning, high-quality learning resources have become much more accessible. This increased accessibility comes in the form of large class size as in MOOCs or frequent reuse of course archives as in colleges, both of which point to the need for adaptive personalization within these online learning environments. On the other hand, learners' interaction with course platforms have left large volumes

of fine-grained behavioral data, paving the way for data-driven adaptivity. Therefore, research on ways to effectively utilize student log data towards automated adaptivity is crucial to educational practice.

One line of research within this field builds adaptivity upon navigational pathways of learners. Compared to earlier studies that are based on cognitive models, this approach makes full use of the latent information within students' navigational sequences and relies less on explicit knowledge about learner behavior. Because the understanding of student online behavior in education research is still at its early stage, this approach has the dual potential to achieve better adaptivity and to add to existing knowledge.

Multiple machine learning algorithms have been leveraged to work towards navigation-based adaptivity. Recent work has started to exploit the power of neural networks. Some papers employ representation learning to embed student behavior for the sake of later stages of recommendation (e.g. [2, 7]), while others step further to construct neural networks for prediction (e.g. [3, 7]). These works have largely focused on MOOCs due to their large data sets and diverse learning pathways which facilitate more effective network training. However, these approaches need to be further validated within other common types of learning environments.

Our work aims to investigate the potential of neural networks to improve teaching and learning in the context of a for-credit online course in a four-year institution. Making an analogy between student navigation and documents, we first embed course URLs to vector space to understand their structures exhibited in student behavior. Then, we use recurrent neural networks (RNN) to predict the next URL for each student based on their previous URL sequence, which can potentially serve as a real-time study guide within the learning management system.

MODELING STRATEGY

In an online course, each student i has their activity sequence: $\{x_1^i, x_2^i, \dots, x_t^i, \dots, x_{T_i}^i\}$, where x_t^i is the t^{th} URL that student i visits during the course and each student visits a total of T_i URLs. A student can visit the same URL multiple times, so the sequence can contain duplicate items. Because students access course pages in a reasonable order, we assume the URL sequences are analogous to word sequences in documents. We will first learn a vector representation of each URL and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

L@S 2018, June 26–28, 2018, London, United Kingdom

© 2018 ACM. ISBN 978-1-4503-5886-6/18/06... 15.00

DOI: <https://doi.org/10.1145/3231644.3231702>

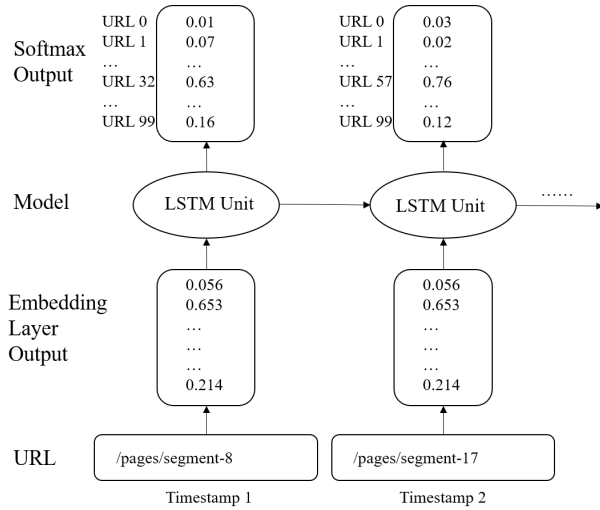


Figure 1. Modeling architecture

visualize all these vectors to gain insights into their relationships. Then, we will feed the learnt representations into RNN architecture to perform next-step prediction.

Vector Representation

We are motivated by the skip-gram model [5] and the prior work of [6] to represent student navigation. The skip-gram model can learn a vector representation of a word (URL) based on its contexts within a large corpus. It's a simple feed-forward, three-layer neural network, which has an input layer, a hidden layer and an output layer. The input is a one-hot vector of a single URL in the sequence. The output is the probability distribution that each URL in the corpus is in the proximity of the input URL. By minimizing cross-entropy loss across all students' sequences, the hidden layer is obtained, which is just the vector representation of the input URL. In the resulting vector space, semantically similar words are close to each other. Two hyperparameters to tune are the window size and the vector dimension.

To visualize the learnt vectors, we use t-SNE algorithm [4]. t-SNE is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional data. In our case, it reduces each vector to two dimensions, and we are then able to plot these vectors on a 2-d plane.

Next-Step Prediction

Because our goal is predicting the next URL given the current URL sequence at any time, RNN architecture would be the best fit. Standard RNNs, however, may suffer the gradient vanishing problem. Therefore, we employ a variant – the long short-term memory (LSTM) model [1] – which solves that issue by adding gating logic to control whether, when and how much to forget information about the past. The update

equations for LSTM are:

$$\begin{aligned}
 f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1}) \\
 i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1}) \\
 C'_t &= \tanh(W_{Cx}x_t + W_{Ch}h_{t-1}) \\
 C_t &= f_t \times C_{t-1} + i_t \times C'_t \\
 o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1}) \\
 h_t &= o_t \times \tanh(C_t)
 \end{aligned} \tag{1}$$

where i_t is the input gate, which controls how much current cell matters; f_t is the forget gate, which controls how much to forget about past; o_t is the output gate, which controls how much the memory cell C_t is exposed to hidden state h_t . C_t and C'_t are intermediary memory cells that are used to update the next hidden state h_t .

As mentioned above, we map the original URL sequences to their vector representation learnt from the skip-gram model, and then feed these embedded sequences to the input layer of LSTM. Output of the LSTM unit will go through a softmax layer and the predicted probability distribution over all URLs is generated. By default, there is only one layer of LSTM unit at each time point, but we also try multiple layers in the experiment part. The overall architecture which is largely inspired by [7] is illustrated in Figure 1.

DATA SET

We use the clickstream data of a large online undergraduate course collected from Canvas system. The course is taught during a normal academic quarter (instead of summer) and lasts for ten weeks. There are four modules throughout the course, each containing a number of segments. Within each segment, there is a lecture video, a short summary of the video, some interactive example problems, and homework that is linked to an external website. At the end of each module, there is a required quiz with a hard deadline. Students are expected to sequentially walk through the four modules, spending two weeks for each module. The quiz deadlines are set according to this expected schedule. Except for the quizzes, other pages can be accessed at any time throughout the course, which allows for variations in student navigation.

A total of 321 students enroll in the course. The total number of clicks (page visits) is 81,678. After cleaning the raw, noisy URLs in the clickstream, there are 96 unique pages, including both core content (lecture videos, assignments, quizzes) and supportive pages (welcome page, discussion session, announcement, etc). We give each page an integer identifier, and these integers constitute the corpus for our models.

RESULTS

Figure 2 visualizes the learnt vector representation for the URLs in the corpus. Each dot represents a URL, and is colored according to the module it belongs to. According to the nature of skip-gram model, if students exactly follow the designed course sequence, vectors for same-module pages should be close to each other. From the figure we can see that this is indeed the case. Given that the course is for-credit and enrolled by full-time college students, this adherence is not unexpected. However, when we break down the students into three grade

Table 1. Next-step prediction results with different models and hyperparameters. $epoch = 10, test_size = 0.16$

model	# layer	# hidden unit per layer	maxlen	learning rate	test accuracy	training accuracy
LSTM	1	32	20	0.005	0.443	0.4395
	1	64	20	0.005	0.481	0.494
	2	32	20	0.005	0.482	0.495
	2	64	50	0.005	0.472	0.483
	2	32	20	0.003	0.508	0.519
GRU	2	32	20	0.003	0.495	0.503
Standard RNN	2	32	20	0.003	0.458	0.467
Baseline (next most common)	/				0.411	0.403

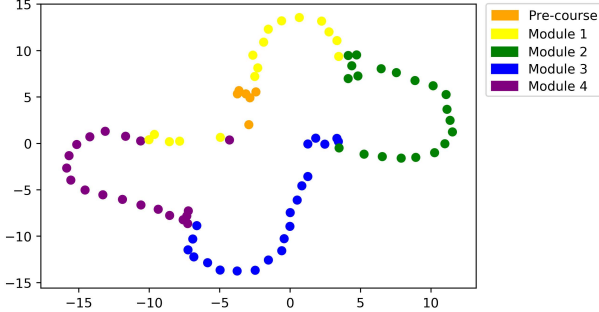


Figure 2. Visualization of vector embedding of course pages. $win_size = 5, vec_dim = 12$

Table 2. Next-step prediction results in different grade groups. Only best results are presented.

grade group	highest test accuracy	baseline (NMC)
A	0.498	0.392
B	0.499	0.407
C/D/F	0.457	0.423

groups and run the same model separately, the results start to diverge, as shown in Figure 3. From A-students to C/D-students, there are more “outliers” in the graph, indicating an increasing tendency to deviate from the designed course structure. This pattern echos the results of a recent paper that investigates a more complicated undergraduate course [6]. It may be because low-performing students are feeling lost in the course content or they are simply playing with the course and not taking it seriously, although the true story needs further investigation.

For the next-step prediction, several hyperparameters of the LSTM model are tuned and the highest test accuracy is 0.508, as shown in Table 1. For comparison, we first calculate the baseline accuracy, where the URL that most commonly follows the current URL in the training data is predicted [8]. From the last row of the table, we can see that the LSTM architecture we use increases prediction accuracy by nearly 10% compared to the baseline. We also conduct experiments on other architecture including standard RNN and gated recurrent unit (GRU). The results show that LSTM outperforms standard RNN to a considerable extent and performs slightly better than GRU. The hyperparameter *maxlen* indicates the number of previous URLs based on which the model predicts

the next-step URL. Compared to the results of similar methods in the context of MOOCs [7, 8], our performance gap between LSTM and the baseline (next most common) is much larger, implying the benefit of neural network methods in this particular scenario. On the other hand, the absolute accuracy is approximately 10% lower than in MOOCs. This may be attributed either to the large volumes of MOOC data leading to less sensitivity to noise, or to the more linear organization of MOOC content via heavy use of navigation buttons and therefore higher predictability.

We also compare the prediction results in different grade groups after separately training and testing the three sub-datasets. From table 2 we can see that student with better grades have a higher prediction accuracy, which may corroborate our previous inference that they are more likely to follow the designed sequence. Also, the LSTM-based results are much better than the baseline. Due to the comparatively small sample size, however, this result is less robust and needs further validation.

DISCUSSION AND FUTURE WORK

This work extends the application of neural network methods in online learning environments. Specifically, it validates the recent literature that works neural networks into MOOCs within a for-credit online college course. Skip-gram models are able to learn meaningful representation of student activity sequences, where we find that students with different grade outcomes diverge in terms of their navigational patterns. Also, using LTSM architecture, the next-step prediction accuracy is much higher than that of predicting the next most common URL and reaches over 50%.

These affordances of neural network methods have practical implications. First, representing student navigation can help instructors better understand students’ learning processes, and tailor their course design to the needs of different students. In our case, low-performing students deserve special modification of the course structure. Second, navigation-based prediction can be useful to construct pipelines of automated instructional support, such as real-time next-step recommendation that has been experimented in MOOCs.

The results in this work are preliminary at best, and there are several directions of future work. First, the course we analyze has standard organizational structures in higher education settings, so we are going to validate the models on other online courses in the same institution. Second, we will be

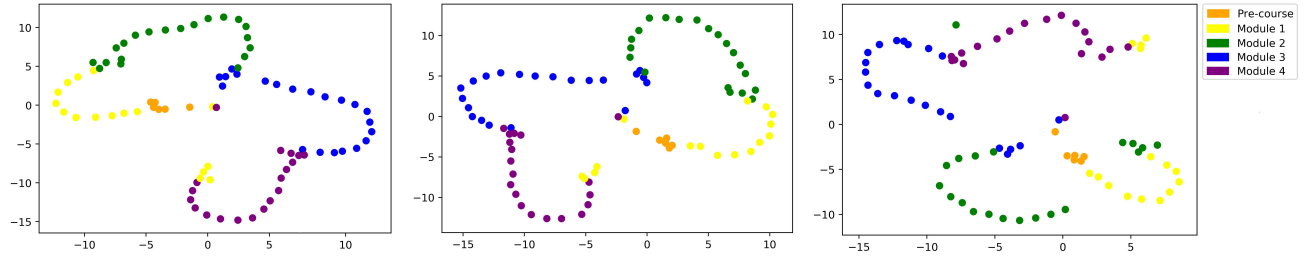


Figure 3. Vector embedding of course pages in different grade groups. Left: A-students; middle: B-students; right: C/D- students. $win_size = 5, vec_dim = 12$

taking into account time on page as inter-URL distance in both representation and prediction processes. Third, using similar strategies we will learn student representations to facilitate the understanding and evaluation of individual students.

ACKNOWLEDGMENTS

This paper is based upon work supported by the National Science Foundation under Grants Number 1535300. We would like to thank Qiujie Li and Jihyun Park for their assistance in acquiring the data used in this paper.

REFERENCES

1. Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
2. Severin Klingler, Rafael Wampfler, Tanja Käser, Barbara Solenthaler, and Markus Gross. 2017. Efficient Feature Embeddings for Student Classification with Variational Auto-encoders. In *Proceedings of the 10th International Conference on Educational Data Mining*. EDM, 72–79.
3. Yuntao Li, Chengzhen Fu, and Yan Zhang. 2017. When and who at risk? Call back at these critical points. In *Proceedings of the 10th International Conference on Educational Data Mining*. EDM, 168–173.
4. Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.
5. Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 746–751.
6. Zachary A. Pardos and Lev Horodyskyj. 2017. Analysis of Student Behaviour in Habitable Worlds Using Continuous Representation Visualization. *ArXiv e-prints* (Oct. 2017).
7. Zachary A. Pardos, Steven Tang, Daniel Davis, and Christopher Vu Le. 2017. Enabling Real-Time Adaptivity in MOOCs with a Personalized Next-Step Recommendation Framework. In *Proceedings of the Fourth ACM Conference on Learning @ Scale - L@S '17*. ACM, Cambridge, MA, USA, 23–32.
8. Steven Tang and Zachary A. Pardos. 2017. Personalized Behavior Recommendation : A Case Study of Applicability to 13 Courses on edX. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, Bratislava, Slovakia, 165–170.