A Deep Model for Predicting Online Course Performance

Hamid Karimi[1]* **Jiangtao Huang**[2]*, **Tyler Derr**[1]
[1]Data Science and Engineering Lab, Michigan state University, {karimiha, derrtyle}@msu.edu
[2]School of Computer and Information Engineering, Nanning Normal University, China, hjt@gxtc.edu.cn

Abstract

Online learning has attracted a large number of participants because it has no limit to enrollment and regardless of personal background and location. One of main goals of education is improving students' learning gain. However, the completion rates for online learning are notoriously low. We focus on predicting students' learning performance early and help instructors to provide intervention in-time. We propose a deep online learning performance prediction model incorporate clickstream and demographic data of students. The experiments on the Open University Learning Analytics Dataset (OULAD) show that fusion of learner demographic information can make up for inadequate online learning behavior data early and improve prediction performance. And our model can achieve reliable performance both in intra-course and inter-course outcome prediction.

Introduction

Online learning provides lecture videos, online assessments, discussion forums, and even live video discussions via the internet. Its environments have presented convenient learning opportunities and enormous learning resources for various types of participants from all over the world. Different from tradition brick-and-mortar based education, online learning breaks the boundaries of time, space and educational resources. At the same time, many educational institutions (including top universities) provide online courses and they can enroll a significant number of students compared to traditional in-person education. Furthermore, people of all ages, cognitive backgrounds and education levels can participate in these courses. This therefore promotes a new form of education development. Most importantly, the cost of online learning compared to traditional education is significantly low. Hence, in recent years, online learning is booming and has attracted a large number of students.

In general, the main goal of education is paying close attention to students and improving their learning gain. In traditional education, students meet in person with instructors in a classroom setting. Teachers can interact with the students and can comprehensively assess their performance from multiple perspectives, such as cognition, mentality, sentiment, and so on. Consequently, they can take actions to provide intervention in a timely manner as illustrated in the right side of Figure 1. In this way, it ultimately can lead to less students dropping or failing the course. However, in online learning systems the situation is different. The inherent less interactions between students and the instructors, high student-teacher ratio, and student diversity cause teachers, if any, not to timely and comprehensively evaluate the learning gain of each student through the online learning platform. Therefore, in online learning, dropout and failure rates are higher than traditional educational systems. For instance, currently, the completion rates of Massive Open Online Courses (MOOCs), an extension of online learning technologies, are low (0.7%-52.1%, with a median value of 12.6% reported by (Jordan 2015)). The same situation happens in other online courses from universities like Open University UK and China (Jha, Ghergulescu, and Moldovan 2019). We should emphasis that online learning is an independent social network, and we face many challenges if wanting to lower the dropout and failure rates. The first major challenge is that students in online learning environments primarily interact with the system/application instead of the instructor as demonstrated in Figure 1 (left side), so we need a way to automatically make assessments about the students progress. To achieve this, we need to have a predictive system automatically predicting the outcome of each student's performance in the course e.g., fail or pass. This ultimately allows the system to provide additional attention to these students to help alleviate the dropout and failure rates.

In online learning, the platform has limited knowledge of students. Typically, these platform only have access to student profiles (i.e., demographic data) and furthermore can log students' interactions with the platform, which is typically in the form of click behaviors. Furthermore, due to the fact that students typically drop out early in the courses (75 percent of dropouts occur in the first weeks as reported in (Santos et al. 2014)), the platform is desired to detect which student is likely to dropout (or fail) as early as possible to offer interventive measures to hopefully prevent these negative outcomes. In addition, unlike a human instructor that

*Co-first author

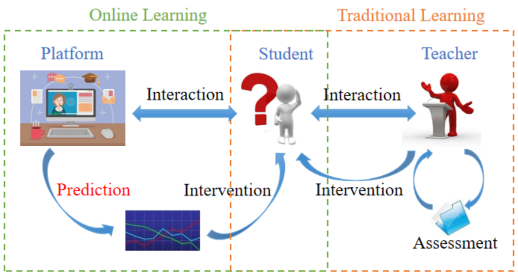


Figure 1: Visual comparison of the learning/intervention process between online and in person education systems

can use multiple assessment strategies, complicated evaluations, and perhaps even sometimes subjective intuitions, the online learning platform will need a quantifiable measurement to determine the learning performance of each student. Hence, the focus of this paper is an intelligent model utilizing student-system interactions and/or demographic data, if applicable, to predict the student's performance outcome in a course. Next we briefly describe the proposed approach.

We propose to harness the power of deep learning to train a model based on a past course such that we can utilize this trained model to evaluate the students in real-time in future courses (including at the early stages of the course). For the systems assessment of the students learning performance, we use the prediction of the course grade as a proxy; in other words, our model seeks to make the prediction of the final outcome of the course for each student. The input to the model is click behaviors which are inherently temporal streams and are highly dynamic. Therefore, to extract salient features from the temporal click data, we utilize Long Short-Term Memory (Hochreiter and Schmidhuber 1997) that can effectively capture temporal dependencies in the sequential click data. At early weeks of a course, however, there exist limited click data that can make the model not perform acceptably. To address this challenge, we further jointly utilize the student demographic data harnessing another form of a deep neural network, namely fully connected layers. The main contributions of this work are as follow.

- We propose a deep model for predicting student learning performance by jointly using student click behaviors considering temporal information and student demographics to assist in cold-start predictions near the course start.
- We perform comprehensive experiments for predicting student performance across different online course domains, perform feature analysis between the click behavior and student demographic data, and the robustness of the learned patterns when even predicting across domains.

Problem Statement

In this section, we introduce some mathematical notations and formally define the given problem.

Suppose from the set of courses in an online system we have a subset of m courses denoted as $\mathcal{C} = \{ c_1, c_2 \cdots c_m\}$. Furthermore, let there be n students in the online system having enrolled in at least one of the m courses in $\mathcal{C}$, which we denote as $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$. For each of the students s_i the system will have collected some demographic information that can be represented as the vector $\mathbf{d}_i \in \mathbb{R}^d$ with d being the dimension size after having encoded the demographic data. In addition to the demographic data, the system is assumed to have the collected the clicking behavior for each student s_i enrolled in course c_j that we represent as $Q_{ij} = \{\mathbf{q}_{ij}^1, \mathbf{q}_{ij}^2, \cdots, \mathbf{q}_{ij}^{K_j}\}$ where $\mathbf{q}_{ij}^w$ represents an encoding of the clicking behavior of student s_i during the w^{th} week of course c_j and K_j represents the total number of weeks in course c_j. For each student s_i we represent their performance outcome in course c_j as o_{ij}, where we assume there can be p outcomes, such as "passed" or "failed". Now, given the notations listed above, we seek to learn a model $f(.,.|\theta)$ having parameters θ such that it can predict the course student outcomes $\mathcal{O}$ as follows:

$$T(\mathcal{Q}^k, \mathcal{D}, \mathcal{O}, f(.,.|\theta)) \rightarrow \hat{\theta} \tag{1}$$

where we use T to denote the learning process, $\mathcal{Q}^k$ is used to represent the click data for a given set of courses $\mathcal{C}$ using only the first k weeks of data, $\mathcal{D}$ denotes the set of demographic data for the students in $\mathcal{S}$, $\mathcal{O}$ represents the performance outcomes of the students in $\mathcal{S}$ and the learned parameters of $f(.,.|\theta)$ are given by $\hat{\theta}$. Then, we can later use our trained model $f(.,.|\hat{\theta})$ as follows for making the outcome predictions $\bar{\mathcal{O}}$ on a new set of courses $\bar{\mathcal{C}}$ with the associated enrolled students $\bar{\mathcal{S}}$ having click data $\bar{\mathcal{Q}}^k$ and demographic data $\bar{\mathcal{D}}$

$$f(\bar{\mathcal{Q}}^k, \bar{\mathcal{D}}|\hat{\theta}) \rightarrow \bar{\mathcal{O}} \tag{2}$$

Now, given the formal definition of the problem for predicting student learning outcomes we introduce our proposed model for $f(.,.|\theta)$.

Proposed Model

We aim to learn a model that can perform early predictions on the outcome of students enrolled in a given course. The motivation for this ability is shown in Figure 1 where a prediction can then be harnessed to perform innervation to students that are predicted as likely to not perform well in the course. To achieve this, we proposed a Deep Online Performance Prediction model illustrated in Figure 2 described in the following.

To achieve a reasonable performance for predicting a course outcome, we must overcome the challenge of how to incorporate the click data which is inherently a sequential stream. A naive way is to concatenate all weekly clicks into a single vector. Doing so, however, fails to capture temporal information manifested in interaction of students with online application. Hence, to better represent the click data, we utilize Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997). LSTM is an effective variant of Recurrent Neural Networks which has been designed to extract temporal features from sequential data e.g., videos (Srivastava, Mansimov, and Salakhudinov 2015) , speech (Graves, Jaitly, and Mohamed 2013), text (Lee and Dernoncourt 2016), as on on.

We fix the length of click sequence for all students to be K (e.g., 10 weeks). Then for a given the sequential data Q_{ij}, at each week $t \in [1, K]$, an LSTM unit takes the t-th week's click feature vector $\mathbf{q}_{ij}^t$ as the input and uses Eq. (3) to produce the output vector h_t.

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \mathcal{G}(W_c x_t + U_c h_{t-1} + b_c) \\ h_t &= o_t \circ \mathcal{G}(c_t) \end{aligned} \tag{3}$$

where i_t, f_t, o_t, h_t, and c_t denote input gate vector, forget gate vector, output gate vector, LSTM output unit vector, and memory cell vector, respectively. W, U, and b are LSTM parameters. $\sigma(.)$ denotes the sigmoid function, $\mathcal{G}$ is an a non-linear activation function, and $\circ$ indicates the pointwise product operator. The final output of the LSTM h_K i.e., output of last LSTM unit.

In addition to click data we incorporate demographic data of students in the model. To this end, we utilize a fully connected layer shown as follows.

$$f_d = \mathcal{G}(\mathbf{W} \times \mathbf{d_i} + \mathbf{b}) \tag{4}$$

where $\mathcal{G}$ is a non-linear activation function, $\mathbf{W}$ is some weight matrix, and $\mathbf{b}$ is a bias vector.

Then, as shown in Figure 2, we concatenate h_K and f_d and pass to a classifier. The classifier first maps the merged vector to a p-size vector (we have p different outcomes) and then we use the softamx to get the outcome probabilities.

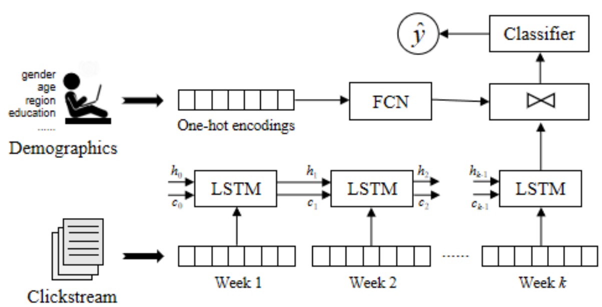


Figure 2: The proposed Deep Online Performance Prediction model (DOPP)

Experiments

In this section, we conduct some experiments to verify the working of our proposed method and compare it with baselines. First, we describe the dataset and then will explain the experimental settings along with baseline methods. Finally in this section, we present the experimental results as well as discussions.

Dataset

Online education platforms utilize virtual learning environments (VLEs) to collect records about all students' interactions and provide the opportunity for analysing students' learning behavior. In this study, we use that data of The Open University Learning Analytics Dataset (OULAD) (Kuzilek, Hlosta, and Zdrahal 2017), which contains information of 22 open university courses for years 2013 and 2014 and 32,593 students. The dataset includes both student demographic information, student assessment results and daily interactions with the university's VLEs (10,655,280 entries). We use OULAD and select one social science course (module name is 'BBB') and two Science, Technology, Engineering and Mathematics (STEM) courses (module names are 'DDD' and 'FFF'). Table 1 shows the descript of the dataset. Note that a portion of registered students do not participate in the course and thus have no interaction with the online platform. Therefore, in this work, we only consider those participating students who have learning behavior log in our analysis. As shown in the last column of Table 1, the outcome of a course for a student can have four different categories including *Distinction* (**D**), *Pass* (**P**), *Fail* (**F**), and *Withdrawn* (**W**).

Table 1: The description of the dataset

Name	**Domain**	**Period**	**# Students**	**# Weeks**	**Outcomes (D, P, F, W)**
BBB	Social Science	2013B	1,537	35	(10.08%, 42.16%, 26.94%, 20.82%)
BBB	Social Science	2013J	1,870	39	(9.41%, 47.91%, 25.24%, 17.43%)
BBB	Social Science	2014B	1,294	34	(12.83%, 43.35%, 27.90%, 15.92%)
BBB	Social Science	2014J	1,921	38	(9.37%, 50.49%, 19.16%, 20.98%)
DDD	STEM	2013B	1,214	35	(4.45%, 37.56%, 28.58%, 29.41%)
DDD	STEM	2013J	1,768	38	(5.54%, 41.35%, 23.42%, 29.69%)
DDD	STEM	2014B	1,116	35	(10.66%, 32.26%, 22.31%, 34.77%)
DDD	STEM	2014J	1,647	38	(6.80%, 41.29%, 21.74%, 30.17%)
FFF	STEM	2013B	1,510	35	(7.81%, 43.97%, 27.28%, 20.93%)
FFF	STEM	2013J	2,098	39	(8.91%, 43.28%, 24.12%, 23.69%)
FFF	STEM	2014B	1,363	35	(7.85%, 40.13%, 27.29%, 24.72%)
FFF	STEM	2014J	2,121	39	(12.16%, 40.50%, 18.20%, 29.14%)

Input features. To represent click features we simply count the different number of weekly clicks a student make e.g., accessing resources, web-page click, forum click, quiz attempt, and so on. The size of weekly click vector ($\mathbf{q}_{ij}^t$) is 20. In addition, we use different demographic information such as gender, age, highest education level, and so on. The vector size of input demographic feature for a student is 36. We use one-hot encoding to represent it.

Experimental Settings

In this paper, we analyse for online course performance prediction for two settings, namely binary classification and 4-class classification. For the former *Pass* and *Distinction* are considered as *Pass* while *Fail* and *Withdrawn* are consider as *Fail*. For 4-class classification, obviously, every outcome is considered as a class. We use data of year 2013 for training and 2014 for testing (all three courses). We use the 20% of the training data as a validation set to tune the hyperparameters. Using the development set we found the best model architecture for binary classification having 50 neurons for LSTM hidden size and a one layer FCN with 100 neurons. For 4-class classification the best model has 100 hiden size for LSTM and a one-layer FCN with 50 hidden neurons. The implementation is done using PyTorch package[1]. Each simulation is run for 5000 steps with learning rate is set to 0.001 and decaying rate 0.99 every 100 steps. Batch size is set to 100 samples (students) and use backpropagation to tune the model parameters. As for evaluation metric we use F1-score which is harmonic mean of recall and precision[2].

Results and Discussions

We compare the proposed method with the following baseline methods.

- SVM. We train a support vecor machine with radial basis function (RBF) kernel. We utilize scikit-learn Python package[3] with default settings.
- Logistic regression. We train a logistic regression model using, again, scikit-learn package with default settings.
- DOPPFCN. This baseline is a variant of the proposed model. Instead of a LSTM component we concatenate all weekly data and pass it to a one-layer fully connected network with 100 hidden neurons.

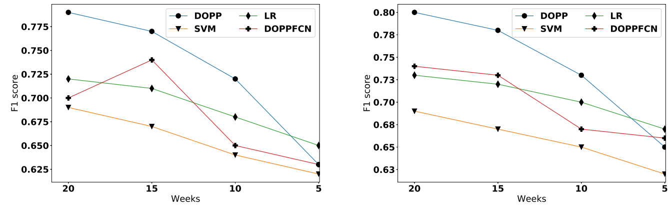
Binary Classification Figures 3, 4, and 5 show the binary classification for courses BBB, DDD, and FFF, respectively. We train the models on a course for periods 2013B,2013J and test it on the same course for periods 2014B, 2014J. We evaluate the methods for different number of weekly click data i.e., 20,15,10, and 5 weeks. Based on the results in these figures we make the following observations:

- In all courses, the more weekly click data is introduced, the better we methods can predict the students' course outcome. DOPP, however, compared to the baselines, enjoys more of such performance increase. In particular, as early as 20 weeks from the start of a course (i.e., almost at the middle of the a course according to Table 1) with a very high accuracy we can predict student's outcome (Fail or Pass). This allows teachers or online course administration to take actionable and interventive measures to help students with poor performance.
- Adding demographic data boosts the model performance. In particular, demographic data an auxiliary source is helpful when just 5 weeks click data is used. Note, again DOPP enjoys more from this boost. The fact that students' demographics information positively affect a course outcome prediction accuracy has an important message: all things being equal, not all people perform equally. Perhaps this can help online course designer to adopt course content or presentation according to a student's demographic data e.g., their education background.
- DOPP achives a better performance than DOPPFCN. This shows the fact the LSTM component as a machinery extracting temporal features from click behaviours is necessary and affect the model's predictive power.

[1]https://pytorch.org/

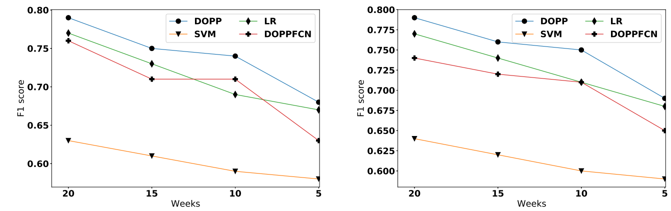
[2]https://en.wikipedia.org/wiki/F1_score

[3]https://scikit-learn.org



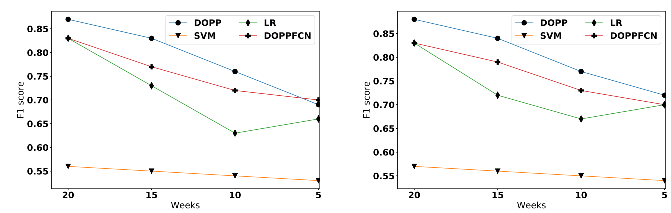
(a) Click (b) Click+Demographics

Figure 3: Performance evaluation for binary classification and course BBB. Random F1 score is 0.39



(a) Click (b) Click+Demographics

Figure 4: Performance evaluation for binary classification and course DDD. Random F1 score is 0.38



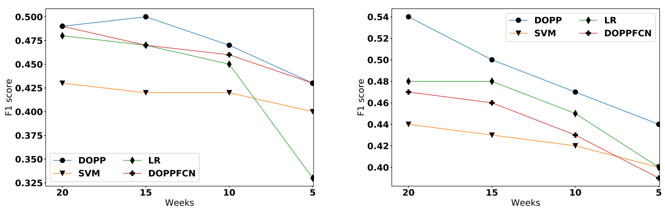
(a) Click (b) Click+Demographics

Figure 5: Performance evaluation for binary classification and course FFF. Random F1 score is 0.40

Four-class Classification

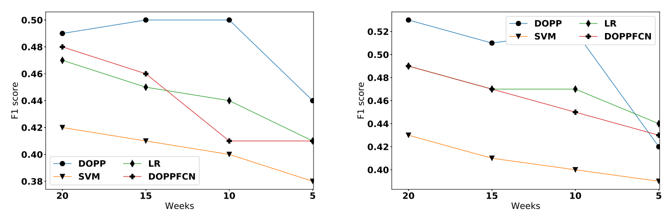
In this part, we show the results for 4-class classification problem. Figures 6, 7, and 8 show 4-class classification for courses BBB, DDD, and FFF, respectively. We make the following observations based on these results.

- The observations we made for binary classification hold for 4-class classifications. In particular, DOPP still outperforms baseline approaches, more weekly click data is helpful in course outcome prediction, and LSTM can effectively handle sequential that than simple concatenation followed by fully connected layer (i.e., DOPPFCN).
- Since more classes are considered, compared to binary classification, 4-class classification is a harder task (see the random performance below each figure). In particular, in 4-class classification *Withdrawn* is considered as a separate class, which might be "conceptally" hard for a model to discern fail from withdrawn.
- In both binary classification and 4-class classification, we observe a poor performance for the proposed model DOPP. This is because of the following. The number of interactions students have with online platform is at the early stages of the course is low. Consequently, DOPP



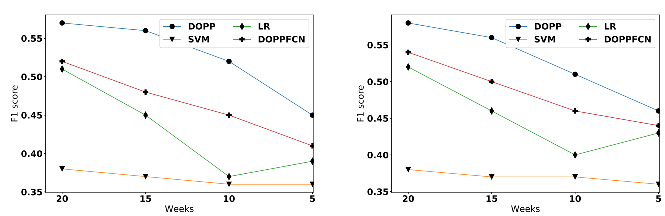
(a) Click (b) Click+Demographics

Figure 6: Performance evaluation for 4-class classification and course BBB. Random F1 score is 0.24



(a) Click (b) Click+Demographics

Figure 7: Performance evaluation for 4-class classification and course DDD. Random F1 score is 0.22



(a) Click (b) Click+Demographics

Figure 8: Performance evaluation for 4-class classification and course FFF. Random F1 score is 0.24

(and as the matter of the fact other models) have less information to decide about a student's outcome. This is in particular challenging for DOPP since it is a deep model approach and requires sufficient data to optimize the parameters.

Intra-course and Inter-course Outcome Evaluation In this section, we study the intra-course and inter-course performance evaluation. For intra-course setting, we train a model on a BBB,DDD, and FFF for 2013B and 2013J periods and test on the same course for 2014B and 2014B. The results presented above all belong to this setting. In addition, we perform an inter-course analysis as well. More specifically, we train the model on a course and test the performance on other courses. For these experiments, we use 20 weeks click data plus demographics. The results for binary classification and 4-class classification are shown, respectively, in Tables 2 and 3. The courses in the rows indicate train courses and those on the column test courses. We make the following observations based on these results.

- Expectedly, when the train course and the test course are the same (i.e., intra-course setting), the model achieves better results. This seems reasonable since clicking patterns are expected for the course in the past (train) and the one in the future (test) and the model can more easily extracts such patterns.
- Although the results for inter-course results are no as good as the ones for intra-course, we still see that the DOPP can effectively achieves reasonable performance. This indicates that proposed model DOPP detects salient click and demographic patters which are transferable from a course to another.
- We can observe that the model obtains good results for intra-domain experiments (i.e., the train and test course are from the same domain, see Table 1). This means that model has extracted transferable features specific to a domain. Although, in general, inter-domain results are not as high as intra-domain ones, we can see the model still acceptable performance specially for models trained on social sciences domain (i.e., BBB) and tested on other domain courses (i.e., DDD and FFF).

Table 2: Intra-course and inter-course binary classification model performance results

	BBB	DDD	FFF
BBB	0.7989	0.7970	0.8615
DDD	0.4561	0.7898	0.6814
FFF	0.6159	0.7417	0.8757

Table 3: Intra-course and inter-course 4-class classification model performance results

	BBB	DDD	FFF
BBB	0.5415	0.5102	0.5129
DDD	0.2639	0.5312	0.5039
FFF	0.4048	0.5029	0.5848

Related Work

Student dropout and performance prediction has become one of the issues of concern in online learning data analysis. In recent year, Many researchers have focused on this issue and applied various machine learning methods to conduct dropout prediction studies. Because MOOC is a grown rapidly form of online learning in recent years, a lot of work are based on analysis of MOOC data. In (Taylor, Veeramachaneni, and O'Reilly 2014), they extracted 27 interpretive features and then use logistic regression to predict student persistence prediction. The authors of (Ramesh et al. 2014) used probabilistic soft logic to model student survival by constructing probabilistic soft logic rules and associating them. Different from (Ramesh et al. 2014) mainly considering forum features, (Kloft et al. 2014) did not further consider forum data and only make use of clickstream data to train their prediction model which contains a principal component analysis and a linear support vector machine for each week. More comprehensively, (Gardner and Brooks 2018) used standard classification trees and adaptive boosted trees to construct their two-stage Friedman and Nemenyi procedure for drop out prediction by processing different features such as clickstream-based, forum-based and assignment-based features. And in (Chen et al. 2019), the authors studied hybrid method for dropout prediction by combining decision tree and extreme learning machine. An unsupervised method is proposed in (Liao, Tang, and Zhao 2019). They proposed a new similarity calculation method, and then make dropout prediction via clustering and tensor completion.

In addition to traditional machine learning methods, some researchers have tried to use different deep learning models for dropout prediction of online courses. (Fei and Yeung 2015) used recurrent neural network model with long short-term memory cells to deal with the features extracted from students' interaction with lecture videos, forum, quiz or problem, and so on. (Whitehill et al. 2017) explored the potential benefits of employing a deep, fully-connected, feed-forward neural network for dropout prediction. Different from previous work, (Feng, Tang, and Liu 2019) proposed a context-aware feature interaction network to incorporate context information, including participant and course information. And they used a attention-based mechanism for learning activity features by using context information. Some models perform well, but they are basically analyzed learning behavior data in the same course.

Conclusion

In a large-scale open education environment, timely prediction of student learning gain is a necessary way to help instructors in providing effective intervention. Online learning platforms have collected fine-grained learning behavior data, which providing opportunity to study prediction methods. However, the diversity of students and the sparseness of data lead challenges of improving prediction performance. At the same time, in order to ensure the effectiveness of the intervention, the prediction model needs to detect at-risk student early. We consider these problem and propose a deep model which combining online learning behavior with student demographics. And we test the proposed model on the OULAD dataset. The results show the feasibility of the model. And it can predict at-risk students of on-going courses by the model trained from courses of the same domain. Different from other methods for predictive analysis in the same course, this may be exactly the way needed.

In future, we will analysis the imbalance and sparse issues of the dataset. Further optimize and improve model performance. And we will study how to construct a generalized model that can utilize historical data from all courses and semesters through introducing the learning of embeddings for a specific course and semester pair. One direction for this can be to harness graph neural networks for the heterogeneous network consisting of students and courses as nodes with their resulting outcome representing the edge type between the two nodes.

References

Chen, J.; Feng, J.; Sun, X.; Wu, N.; Yang, Z.; and Chen, S. 2019. Mooc dropout prediction using a hybrid algorithm based on decision tree and extreme learning machine. *Mathematical Problems in Engineering* 2019.

Fei, M., and Yeung, D.-Y. 2015. Temporal models for predicting student dropout in massive open online courses. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 256–263. IEEE.

Feng, W.; Tang, J.; and Liu, T. X. 2019. Understanding dropouts in moocs. *Association for the Advancement of Artificial Intelligence*.

Gardner, J., and Brooks, C. 2018. Dropout model evaluation in moocs. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Graves, A.; Jaitly, N.; and Mohamed, A.-r. 2013. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, 273–278. IEEE.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Jha, N.; Ghergulescu, I.; and Moldovan, A.-N. 2019. Oulad mooc dropout and result prediction using ensemble, deep learning and regression techniques.

Jordan, K. 2015. Massive open online course completion rates revisited: Assessment, length and attrition. *International Review of Research in Open and Distance Learning* 16(3).

Kloft, M.; Stiehler, F.; Zheng, Z.; and Pinkwart, N. 2014. Predicting mooc dropout over weeks using machine learning methods. In *Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs*, 60–65.

Kuzilek, J.; Hlosta, M.; and Zdrahal, Z. 2017. Open university learning analytics dataset. *Scientific data* 4:170171.

Lee, J. Y., and Dernoncourt, F. 2016. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*.

Liao, J.; Tang, J.; and Zhao, X. 2019. Course drop-out prediction on mooc platform via clustering and tensor completion. *Tsinghua Science and Technology* 24(4):412–422.

Ramesh, A.; Goldwasser, D.; Huang, B.; Daume III, H.; and Getoor, L. 2014. Learning latent engagement patterns of students in online courses. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Santos, J. L.; Klerkx, J.; Duval, E.; Gago, D.; and Rodríguez, L. 2014. Success, activity and drop-outs in moocs an exploratory study on the uned coma courses. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge*, 98–102. ACM.

Srivastava, N.; Mansimov, E.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, 843–852.

Taylor, C.; Veeramachaneni, K.; and O'Reilly, U.-M. 2014. Likely to stop? predicting stopout in massive open online courses. *arXiv preprint arXiv:1408.3382*.

Whitehill, J.; Mohan, K.; Seaton, D.; Rosen, Y.; and Tingley, D. 2017. Delving deeper into mooc student dropout prediction. *arXiv preprint arXiv:1702.06404*.