Understanding Dropouts in MOOCs

Wenzheng Feng,[†] **Jie Tang**[†] and **Tracy Xiao Liu**[‡]
[†]Department of Computer Science and Technology, Tsinghua University
[‡]Department of Economics, School of Economics and Management, Tsinghua University
fwz17@mails.tsinghua.edu.cn, jietang@tsinghua.edu.cn, liuxiao@sem.tsinghua.edu.cn

Abstract

Massive open online courses (MOOCs) have developed rapidly in recent years, and have attracted millions of online users. However, a central challenge is the extremely high dropout rate — a recent report shows that the completion rate in MOOCs is below 5% (Onah, Sinclair, and Boyatt 2014; Kizilcec, Piech, and Schneider 2013; Seaton et al. 2014). Therefore, it is necessary to understand what are the underlying factors that induce the users to drop out and what are the major motivations behind the users' study. In this paper, employing a dataset from XuetangX[1], one of the largest MOOCs in China, we conduct a systematical study for the dropout problem in MOOCs. We found that the users' learning behavior can be clustered into several categories. Our statistics also reveal high *correlation* between dropouts of different courses and strong *influence* between friends' dropout behaviors. Based on the gained insights, we propose a Context-aware Feature Interaction Network (CFIN) to model and to predict users' dropout behavior. CFIN utilizes context-smoothing technique to smooth feature values with different context, and use attention mechanism to combine user and course-context information into the modeling framework. Experiments on two large datasets show that the proposed method achieves better performance than several state-of-the-art methods. The proposed method model has been deployed on a real system to help improve user retention.

Introduction

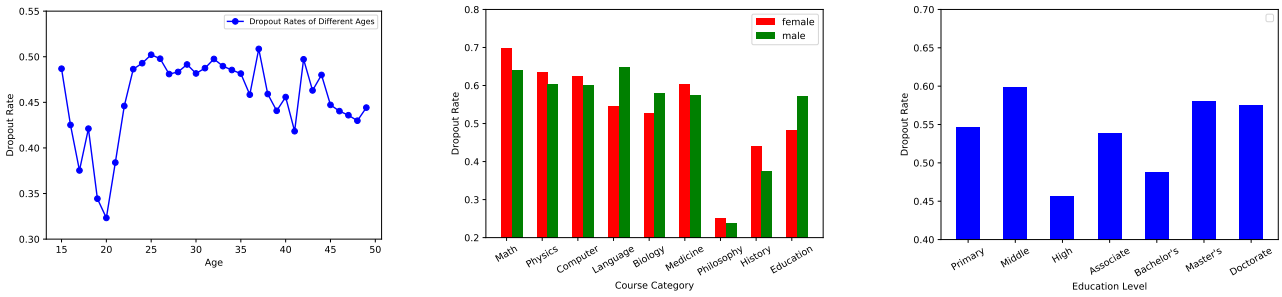
Massive open online courses (MOOCs) have become increasingly popular. Many MOOC platforms have been launched. For example, Coursera, edX, and Udacity are three pioneers, followed by many others from different countries such as XuetangX in China, Khan Academy in North America, Miriada in Spain, Iversity in German, FutureLearn in England, Open2Study in Australia, Fun in France, Veduca in Brazil, and Schoo in Japan (Qiu et al. 2016). By the end of 2017, the MOOC platforms have offered 9,400 courses worldwide and attracted 81,000,000 online registered students (Shah 2018). Recently, a survey from Coursera shows that MOOCs are really beneficial to the learners who *complete* courses, where 61% of survey respondents report MOOCs' education benefits and 72% of those report career benefits (Zhenghao et al. 2015).

However, on the other hand, MOOCs are criticized for the low completion ratio (He et al. 2015). Indeed, the average course completion rate on Edx is only 5% (Kizilcec, Piech, and Schneider 2013; Seaton et al. 2014). We did a similar statistic for 1,000 courses on XuetangX, and resulted in a similar number — 4.5%. Some people argue that the main possible reason for the high dropout rate on MOOCs might be the low monetary cost, i.e., most MOOCs courses are free. Some other people state that motivations of the users to study online very differently. Figure 1 shows several observational analyses. As can be seen, Age is an important factor — young people are more inclined to drop out; Gender is another important factor — roughly, female users are more likely to drop science courses and male users are more likely to give up non-science courses; finally, educational background is also important. This raises several interesting questions: 1) what are the major dropout reasons? 2) what are the deep motivations that drive the users to study or induce them to drop out? 3) is that possible to predict users' dropout behavior in advance, so that the MOOCs platform could deliver some kind of useful interventions (Halawa, Greene, and Mitchell 2014)?

Employing a dataset from XuetangX, the largest MOOC platform in China, we aim to conduct a systematical exploration for the aforementioned questions. We first perform a clustering analysis over users' learning activity data and found that users' studying behavior can be clustered into several categories, which implicitly correspond to different motivations that users study MOOC courses. The analyses also disclose several interesting patterns. For example, there are high correlations between dropout rates of similar courses; friends' dropout behaviors strongly influence each other — the probability that a user drop out from a course increases quickly to 65% when the number of her/his dropout friends increases to 5.

Based on the analyses results, we propose a Context-aware Feature Interaction Network (CFIN) to model and to predict users' dropout behavior. In CFIN, we utilize a context-smoothing technique to smooth values of activity features by using the convolutional neural network (CNN) to fuse the information learned from different sources. Attention mechanisms are then used to combine user and course-

[1]https://xuetangx.com



(a) Age (b) Course Category (c) Education Level

Figure 1: Dropout rates of different demographics of users. (a) user age (b) course category (c) user education level.

context information into the modeling framework. We evaluate the proposed CFIN on two datasets: KDDCUP2015 and XuetangX. The first dataset was used in KDDCUP 2015 and the second one is larger extracted from the XuetangX system. Experiments on both datasets show that the proposed method achieves much better performance than several state-of-the-art methods. We have also deployed the proposed method in XuetangX to help improve user retention.

Related Work

Dropout Prediction. Prior studies apply generalized linear models (including logistic regression and linear SVMs (Kloft et al. 2014; He et al. 2015)) to predict dropout. Balakrishnan et al. (2013) present a hybrid model which combines Hidden Markov Models (HMM) and logistic regression to predict student retention on a single course. Another attempt by Xing et al. (2016) uses an ensemble stacking generalization approach to build robust and accurate prediction models. Deep learning methods are also used for predicting dropout. For example, Fei et al. (2015) tackle this problem from a sequence labeling perspective and apply an RNN based model to predict students' dropout probability. Wang et al. (2017) propose a hybrid deep neural network dropout prediction model by combining the CNN and RNN. Ramesh et al. (2014) develop a probabilistic soft logic (PSL) framework to predict user retention by modeling student engagement types using latent variables. Besides prediction itself, Nagrecha et al. (2017) focus on the interpretability of existing dropout prediction methods. Whitehill et al. (2015) design an online intervention strategy to boost users' callback in MOOCs. Dalipi et al. (2018) review the techniques of dropout prediction and propose several insightful suggestions for this task. What's more, XuetangX has organized the KDDCUP 2015[2] for dropout prediction. In that competition, most teams adopt assembling strategies to improve the prediction performance, and "Intercontinental Ensemble" team get the best performance by assembling over sixty single models.

Engagement Pattern Mining. More recent works mainly focus on analyzing students engagement based on statistical methods and explore how to improve student engagements (Kellogg 2013; Reich 2015). Kizilcec et al. (2013) employ a cluster method to identify the longitudinal engagement trajectories. Anderson et al. (2014) provide a taxonomy for student engagement patterns and study the relationship between student grades and their engagement. Guo et al. (2014) analyze how student engagement pattern varies with different video types. Kim et al. (2014) discover high dropout rate often occurs in long videos, and students who re-watch videos are more likely to drop out than first-time watchers. Zheng et al. (2015) apply the grounded theory to study users' motivations for choosing a course and to understand the reasons that users drop out a course. Qiu et al. (2016) study the relationship between student engagement and their certificate rate, and propose a latent dynamic factor graph (LadFG) to model and predict learning behavior in MOOCs. Chaturvedi et al. (2014) propose a framework to predict instructor's intervention on forums. Ramesh et al. (2015) develop a weakly supervised joint model for aspect-sentiment analysis in forums. Maximilian et al. (2016) use topic models for the psychometric testing of MOOC students based on their forum activities. Utilizing data from a Coursera class, Yang et al. (2013) use a survival model to measure the impact of certain factors on dropout rate. Some other related study could be also found in (Bayer et al. 2012).

Data and Insights

The analysis in this work is performed on two datasets from XuetangX. XuetangX, launched in October 2013, is now one of the largest MOOC platforms in China. It has provided over 1,000 courses and attracted more than 10,000,000 registered users. XuetangX has twelve categories of courses: art, biology, computer science, economics, engineering, foreign language, history, literature, math, philosophy, physics, and social science. Users in XuetangX can choose the learning mode: Instructor-paced Mode (IPM) and Self-paced Mode (SPM). IPM follows the same course schedule as conventional classrooms, while in SPM, one could have more flexible schedule to study online by herself/himself. Usually an IPM course spans over 16 weeks in XuetangX, while an SPM course spans a longer period. Each user can en-

[2]https://biendata.com/competition/kddcup2015

Table 1: Statistics of the KDDCUP dataset.

Category	Type	Number
log	# video activities	1,319,032
	# forum activities	10,763,225
	# assignment activities	2,089,933
	# web page activities	738,0344
enrollment	# total	200,904
	# dropouts	159,223
	# completions	41,681
	# users	112,448
	# courses	39

Table 2: Statistics of the XuetangX dataset.

Category	Type	#IPM*	#SPM*
log	# video activities	50,678,849	38,225,417
	# forum activities	443,554	90,815
	# assignment activities	7,773,245	3,139,558
	# web page activities	9,231,061	5,496,287
enrollment	# total	467,113	218,274
	# dropouts	372,088	205,988
	# completions	95,025	12,286
	# users	254,518	123,719
	# courses	698	515

* #IPM and #SPM respectively stands for the number for the corresponding IPM courses and SPM courses.

roll one or more courses. When one studying a course, the system records multiple types of activities: video watching (watch, stop, and jump), forum discussion (ask questions and replies), assignment completion (with correct/incorrect answers, and reset), and web page clicking (click and close a course page).

Two Datasets. We use two datasets in this study. Both are from XuetangX. The first dataset contains 39 IPM courses and their enrolled students. It was also used for KDDCUP 2015. Table 1 lists statistics of this dataset. With this dataset, we compare our proposed method with existing methods, as the challenge has attracted 821 teams to participate. We refer to this dataset as KDDCUP.

The other dataset contains 698 IPM courses and 515 SPM courses. Table 2 lists the statistics. The dataset contains richer information, which can be used to test the robustness and generalization of the proposed method. This dataset is referred to as XuetangX.

Insights

Before proposing our methodology, we try to gain a better understanding of the users' learning behavior. We first perform a clustering analysis on users' learning activities. To construct the input for the clustering analysis, we define a concept of *temporal code* for each user.

Definition 1. **Temporal Code**: For each user u and one of her enrolled course c, the temporal code is defined as a binary-valued vector $\mathbf{s}_c^u = [s_{c,1}^u, s_{c,2}^u, ..., s_{c,K}^u]$, where $s_{c,k}^u \in \{0, 1\}$ indicates whether user u visits course c in the k-th week. Finally, we concatenate all course-related vectors and generate the temporal code for each user as $\mathbf{S}^u = [\mathbf{s}_{c_1}^u, \mathbf{s}_{c_2}^u, ..., \mathbf{s}_{c_M}^u]$, where M is the number of courses.

Table 3: Results of clustering analysis. C1-C5 — Cluster 1 to 5; CAR — average correct answer ratio.

Category	Type	C1	C2	C3	C4	C5
video	#watch	21.83	46.78	12.03	19.57	**112.1**
	#stop	28.45	68.96	20.21	37.19	**84.15**
	#jump	16.30	16.58	11.44	14.54	**21.39**
forum	#question	0.04	**0.38**	0.02	0.03	0.03
	#answer	0.13	**3.46**	0.13	0.12	0.17
assignment	CAR	0.22	**0.76**	0.19	0.20	0.59
	#revise	0.17	0.02	0.04	**0.78**	0.01
session	seconds	1,715	714	**1,802**	1,764	885
	count	3.61	**8.13**	2.18	4.01	7.78
enrollment	#enrollment	21,048	9,063	**401,123**	25,042	10,837
	total #users	2,735	4,131	**239,302**	4,229	4,121
	dropout rate	0.78	0.29	**0.83**	0.66	0.28

Please note that the temporal code is usually very sparse. We feed the sparse representations of all users' temporal codes into a K-means algorithm. The number of clusters is set to 5 based on a *Silhouette Analysis* (1987) on the data. Table 3 shows the clustering results. It can be seen that both cluster 2 and cluster 5 have low dropout rates, but more interesting thing is that users of cluster 5 seem to be hard workers — with the longest video watching time, while users of cluster 2 seem to be active forum users — the number of questions (or answers) posted by these users is almost $10\times$ higher than the others. This corresponds to different motivations that users come to MOOCs. Some users, e.g., users from cluster 5, use MOOC to seriously study knowledge, while some other users, e.g., cluster 2, may simply want to meet friends of similar interest. Another interesting phenomenon is about users in cluster 4. Their average number of revise answers for assignment (i.e. #reset) is much higher than all the other clusters. Users of this cluster probably are students with difficulties to learn the corresponding courses.

Correlation Between Courses. We further study whether there is a correlation between user's dropout behavior of different/related courses. Specifically, we try to answer the question: will someone's dropout from one course increase/decrease the probability that she/he drops out from another course? We conduct a regression analysis to examine the correlation of a user's dropout behavior between different courses. A user's dropout behavior in a course is encoded as a 16-dim dummy vector, with each element representing whether the user has visited the course in the corresponding week (thus 16 corresponds to the 16 weeks for studying the course). The input and output of the regression model are two dummy vectors which indicate a user's dropout behavior on two different courses in the same semester. By examining the slopes of regression results (Figure 2), we can observe a significantly positive correlation between users' dropout probabilities of different enrolled courses, though overall the correlation decreases over time. Moreover, we did the analysis for courses of the same category and those across different categories. It can be seen

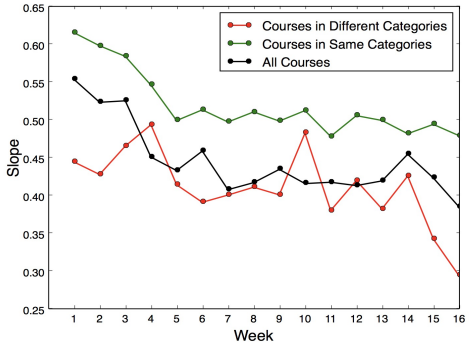


Figure 2: Dropout correlation analysis between courses. The x-axis denotes the weeks from 1 to 16 and the y-axis is the slope of linear regression results for dropout correlation between two different courses. The red line is the result of different category courses, the green line denotes the slope of same category courses, and the black line is pooling results in all courses.

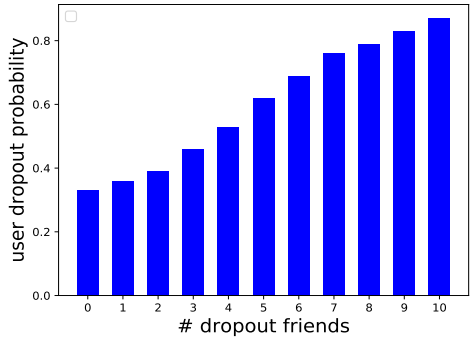


Figure 3: User dropout probability conditioned on the number of dropout friends. x-axis is the number of dropout friends, and y-axis is user's dropout probability.

that the correlation between courses of the same category is higher than courses from different categories. One potential explanation is that when a user has limited time to study MOOC, they may first give up substitutive courses instead of those with complementary knowledge domain.

Influence From Dropout Friends. Users' online study behavior may influence each other (2016). We did another analysis to understand how the influence would matter for dropout prediction. In XuetangX, the friend relationship is implicitly defined using co-learning relationships. More specifically, we use a network-based method to discover users' friend relationships.First, we build up a user-course bipartite graph G_{uc} based on the enrollment relation. The nodes are all users and courses, and the edge between user u and course c represents that u has enrolled in course c. Then we use DeepWalk (2014), an algorithm for learning representations of vertices in a graph, to learn a low dimensional vector for each user node and each course node in G_{uc}. Based on the user-specific representation vectors, we

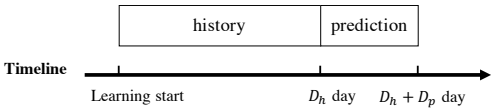


Figure 4: Dropout Prediction Problem. The first D_h days are *history period*, and the next D_p days are *prediction period*.

compute the cosine similarity between users who have enrolled a same course. Finally, those users with high similarity score, i.e., greater than 0.8, are considered as friends.

In order to analyze the influence from dropout friends quantitatively, we calculate users' dropout probabilities conditional on the number of dropout friends. Figure 3 presents the results. We see users' dropout probability increases monotonically from 0.33 to 0.87 when the number of dropout friends ranges from 1 to 10. This indicates that a user's dropout rate is greatly influenced by her/his friends' dropout behavior.

Methodology

We now turn to discuss potential solutions to predict when and whether a user will drop out a specific course, by leveraging the patterns discovered in the above analysis. In summary, we propose an Context-aware Feature Interaction Network (CFIN) to deal with the dropout prediction problem. Different from previous work on this task, the proposed model incorporates context information, including course course correlation and user influence, into a unified framework. Let us begin with a formulation of the problem we are going to address.

Formulation

Our main task is to predict whether a user would dropout from an enrolled course in a pre-specified time window. In order to formulate this problem, we introduce the following definitions.

Definition 2. **Enrollment Relation**:Let $\mathbb{C}$ denote the set of courses, $\mathbb{U}$ denote the set of users, and the pair (u, c) denote user $u \in \mathbb{U}$ enrolls the course $c \in \mathbb{C}$. The set of enrolled courses by u is denoted as $\mathbb{C}_u \subset \mathbb{C}$ and the set of users who have enrolled course c is denoted as $\mathbb{U}_c \subset \mathbb{U}$. We use $\mathbb{E}$ to denote the set of all enrollments, i.e., $\{(u, c)\}$

Definition 3. **Learning Activity**: In MOOCs, user u's learning activities in a course c can be formulated into an m_x-dimensional vector $\mathbf{X}(u, c)$, where each element $x_i(u, c) \in \mathbf{X}(u, c)$ is a continuous feature value associated to u's learning activity in a course c. Those features are extracted from user historical logs, mainly includes the statistics of users' activities.

Definition 4. **Context Information**: Context information in MOOCs comprises user information and course information. User information is represented by user demographics (i.e. gender, age, location, education level) and user cluster. While course information is the course category. The categorical information (e.g. gender, location) is represented by a one-hot vector, while continues information (i.e. age) is represented as the value itself. By concatenating all infor-

Table 4: Average activity feature values on two sample courses — Conversational English and Data Structure. CAR — average correct answer ratio.

Category	Type	Conversational English	Data Structure
video	#watch	74.80	32.48
	#stop	69.53	26.14
	#jump	51.18	14.38
forum	#question	0.049	0.038
	#answer	1.10	0.12
assignment	CAR	0.45	0.41
	#revise	0.09	0.02
session	seconds	1,715	714
	count	3.61	8.13
dropout rate		0.74	0.73

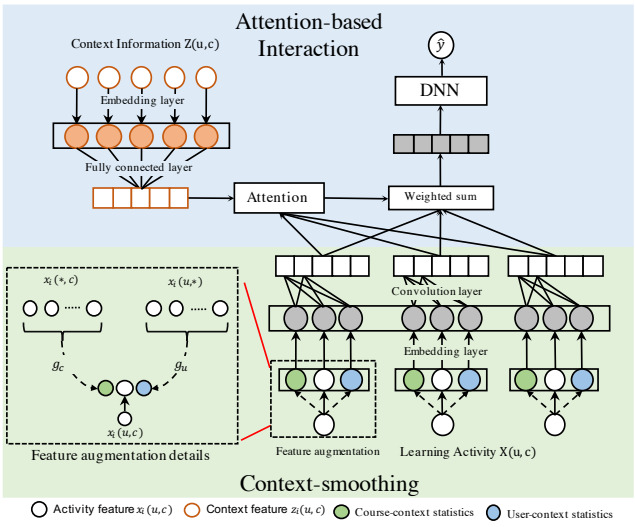


Figure 5: The Architecture of CFIN.

mation representations, the context of a (u, c) pair is represented by a vector $\mathbf{Z}(u, c)$.

With these definitions, our problem of dropout prediction can be defined as: Given user u's learning activity $\mathbf{X}(u, c)$ on course c in *history period* (as shown in Figure 4, it is the first D_h days after the learning starting time), as well as her context information $\mathbf{Z}(u, c)$, our goal is to predict whether u will drop out from c in the *prediction period* (as shown in Figure 4, it is the following D_p days after *history period*). More precisely, let $y_{(u,c)} \in \{0, 1\}$ denote the ground truth of whether u has dropped out, $y_{(u,c)}$ is positive if and only if u has not taken activities on c in the *prediction period*. Then our task is to learn a function:

$$f : (\mathbf{X}(u, c), \mathbf{Z}(u, c)) \rightarrow y_{(u,c)}$$

Please note that we define the prediction of dropouts for all users/courses together, as we need to consider the user and course-context information.

Context-aware Feature Interaction Network

Motivation. From the prior analyses, we find users' activity patterns in MOOCs have a strong correlation with their context (e.g. course correlation and friends influence). Specifically, the value of learning activity vector $\mathbf{X}(u, c)$ is highly sensitive to the context information $\mathbf{Z}(u, c)$. Table 4 presents the average activity feature values on two sample courses of XuetangX (i.e. Conversational English and Data Structure). We observe that these features are totally different across two courses though they have an approximate user dropout rate. To tackle this issue, we employ convolutional neural networks (CNN) to learn a context-aware representation for each activity feature $x_i(u, c)$ by leveraging its context statistics. This strategy is referred to as context-smoothing in this paper. What's more, we also propose an attention mechanism to learn the importances of different activities by incorporating $\mathbf{Z}(u, c)$ into dropout prediction. Figure 5 shows the architecture of the proposed method. In the rest of this section, we will explain the context-smoothing and attention mechanism in details.

Context-Smoothing. The context-smoothing consists of three steps: feature augmentation, embedding and feature fusion. In feature augmentation, each learning activity feature $x_i(u, c) \in \mathbf{X}(u, c)$[3] is expanded with its user and course-context statistics. User-context statistics of feature x_i is defined by a mapping function $g_u(x_i)$ from the original activity feature to several statistics of i^{th} feature across all courses enrolled by u, i.e., $g_u : x_i(u, c) \rightarrow [\text{avg}(\{x_i(u, *)\}), \max(\{x_i(u, *)\}), \ldots]$. While course-context statistics, represented by $g_c(x_i)$, are statistics over all users in course c, i.e., $g_c : x_i(u, c) \rightarrow [\text{avg}(\{x_i(*, c)\}), \max(\{x_i(*, c)\}), \ldots]$. We can define $\hat{\mathbf{X}} = \hat{\mathbf{X}}_g^{(1)} \oplus \hat{\mathbf{X}}_g^{(2)} \oplus \ldots \oplus \hat{\mathbf{X}}_g^{(m_x)}$ to represent augmented activity feature vector, where each $\hat{\mathbf{X}}_g^{(i)} \in \mathbb{R}^{m_g}$ is a feature group which consists of the original feature value x_i and its context statistics: $\hat{\mathbf{X}}_g^{(i)} = [[x_i] \oplus g_u(x_i) \oplus g_c(x_i)]$

Then, similar to most of deep learning methods, each element $\hat{x} \in \hat{\mathbf{X}}$ is converted to a dense vector through an embedding layer. Since $\hat{x}$ is continuous variable, we obtain the corresponding embedding vectors by simply multiplying $\hat{x}$ with a parameter vector $\mathbf{a} \in \mathbb{R}^{d_e}$:

$$\mathbf{e} = \hat{x} \cdot \mathbf{a} \tag{1}$$

By this way, $\hat{\mathbf{X}}$ is projected to an embedding matrix $\mathbf{E}_x \in \mathbb{R}^{m_g m_x \times d_e}$. For easy explanation, we use $\mathbf{E}_g^{(i)} \in \mathbb{R}^{m_g \times d_e}$ to represent the embedding matrix of $\hat{\mathbf{X}}_g^{(i)}$.

After projecting features into embedding vectors, the final step is feature fusion. We employ an one-dimensional convolutional neural network (CNN) to compress $\mathbf{E}_g^{(i)}$ into a vector. The convolution kernel is represented by $\mathbf{W}_{conv} \in \mathbb{R}^{d_f \times m_g d_e}$, which is applied to $\mathbf{E}_g^{(i)}$. More formally, a vector $\mathbf{V}_g^{(i)} \in \mathbb{R}^{d_f}$ is generated from $\mathbf{E}_x^{(i)}$ by

$$\mathbf{V}_g^{(i)} = \sigma(\mathbf{W}_{conv} \delta(\mathbf{E}_g^{(i)}) + \mathbf{b}_{conv})^4 \tag{2}$$

[3]We ommit the notation (u, c) in the following description, if no ambiguity

[4]$\delta(\mathbf{E})$ denotes flatting matrix $\mathbf{E}$ to a vector

Where $\mathbf{b}_{conv} \in \mathbb{R}^{d_f}$ is a bias term. $\sigma(\cdot)$ is activate function. This procedure can be seen as an m_g-stride convolution on $\mathbf{E}_x$. By this way, each feature group $\hat{\mathbf{X}}_g^{(i)}$ is represented by a dense vector $\mathbf{V}_g^{(i)}$. It can be seen as the context-aware representation of the original feature x_i with integrating the context statistics information. We use $\mathbf{V}_x \in \mathbb{R}^{m_x \times d_f}$ to denote the learned feature map of $\mathbf{E}_x$, i.e., $\mathbf{V}_x = [\mathbf{V}_g^{(1)}, \mathbf{V}_g^{(2)}, ..., \mathbf{V}_g^{(m_x)}]^{\mathrm{T}}$.

Attention-based Interaction We now turn to introduce how to learn a dropout probability from activity feature maps $\mathbf{V}_x$ by employing the attention mechanism. First, we need to transform $\mathbf{Z}$ into a dense vector $\mathbf{V}_z \in \mathbb{R}^{d_f}$ through an embedding layer and a fully-connected layer. For continuous features in $\mathbf{Z}$, the embedding strategy is based on Equation 1. While for categorical features, the embedding vector is obtained through multiplying the corresponding one-hot vector by an embedding matrix $\mathbf{W}_{emb}$:

$$\mathbf{e} = \mathbf{z}^{\mathrm{T}} \mathbf{W}_{emb} \tag{3}$$

Where $\mathbf{z} = [0, \cdots, 1, \cdots, 0]^{\mathrm{T}}$ is a sub-vector of $\mathbf{Z}$, representing one categorical feature. We use $\mathbf{E}_z \in \mathbb{R}^{m_z \times d_e}$ to denote the embedding matrix of $\mathbf{Z}$. Then $\mathbf{E}_z$ is fed into a fully-connect network to learn $\mathbf{V}_z$:

$$\mathbf{V}_z = \sigma(\mathbf{W}_{fc}\delta(\mathbf{E}_z) + \mathbf{b}_{fc}) \tag{4}$$

Then $\mathbf{V}_z$ is used to calculate an attention score for $\mathbf{V}_g^{(i)}$:

$$\hat{\lambda}_i = \mathbf{h}_{attn}^{\mathrm{T}} \sigma(\mathbf{W}_{attn}(\mathbf{V}_g^{(i)} \oplus \mathbf{V}_z) + \mathbf{b}_{attn}) \tag{5}$$

$$\lambda_i = \frac{\exp(\hat{\lambda}_i)}{\sum_{1 \le i \le m_x} \exp(\hat{\lambda}_i)} \tag{6}$$

Where $\mathbf{W}_{attn} \in \mathbb{R}^{d_a \times 2d_f}$, $\mathbf{b}_{attn} \in \mathbb{R}^{d_a}$ and $\mathbf{h}_{attn} \in \mathbb{R}^{d_a}$ are parameters. λ_i is the attention score of $\mathbf{V}_g^{(i)}$, witch can be interpreted as the importance of i^{th} feature $x_i \in \mathbf{X}$. We transformed $\mathbf{V}_x$ into a vector $\mathbf{V}_g^{sum}$ by applying a weighted sum on $\mathbf{V}_g^{(i)} \in \mathbf{V}_x$:

$$\mathbf{V}_g^{sum} = \sum_{1 \le i \le m_x} \lambda_i \mathbf{V}_g^{(i)} \tag{7}$$

By this method, the context-aware representations of features in $\mathbf{X}$ are compressed as a vector $\mathbf{V}_g^{sum}$ based on their importance.

In the final step, we feed $\mathbf{V}_g^{sum}$ into an L-layer deep neural network (DNN) to learn the interactions of features. Specifically, the input layer is $\mathbf{V}_g^{sum}$. While each hidden layer can be formulated as:

$$\mathbf{V}_d^{(l+1)} = \sigma(\mathbf{W}_d^{(l)} \mathbf{V}_d^{(l)} + \mathbf{b}_d^{(l)}) \tag{8}$$

where l is the layer depth. $\mathbf{W}_d^{(l)}$, $\mathbf{b}_d^{(l)}$ are model parameters. $\mathbf{V}_d^{(l)}$ is output of l-layer. The final layer a sigmoid function which used to estimate the dropout rate $\hat{y}$:

$$\hat{y}_{(u,c)} = \frac{1}{1 + \exp(-\mathbf{h}_{sig}^{\mathrm{T}} \mathbf{V}_d^{(L-1)})} \tag{9}$$

where $\hat{y}_{(u,c)} \in [0, 1]$ denotes the probability of u dropping out from course c. All the parameters can be learned by minimizing the follow objective function:

$$\begin{aligned} L(\mathbf{\Theta}) = -\sum_{(u,c) \in \mathbb{E}} & [y_{(u,c)} \log(\hat{y}_{(u,c)}) \\ & + (1 - y_{(u,c)}) \log(1 - \hat{y}_{(u,c)})] \end{aligned} \tag{10}$$

where $\mathbf{\Theta}$ denotes the set of model parameters, $y_{(u,c)}$ is the corresponding ground truth, $\mathbb{E}$ is the set of all enrollments.

Model Ensemble

For further improving the prediction performance, we also design an ensemble learning strategy by combining CFIN with the XGBoost (Chen and Guestrin 2016), one of the most effective gradient boosting framework. Specifically, we obtained the output vector of DNN's $(L-1)^{th}$ layer, which is denoted by $\mathbf{V}_d^{(L-1)}$, from the successfully trained CFIN and use it as features to train an XGBoost classifier. The original features, i.e., $\mathbf{X}$ and $\mathbf{Z}$ are also fed to the XGBoost. This strategy is a little like stacking (Wolpert 1992), but the difference is that the latter is to feed one model's predict probability to another one.

Experiments

We conduct various experiments to evaluate the effectiveness of CFIN on the two datasets.

Experimental Setup

Implementation Details. We implement CFIN with tensorflow[5] and adopt Adam (Kingma and Ba 2014) to optimize the model. To prevent overfitting, we apply $L2$ regularization on the weight matrices. We adopt Rectified Linear Unit (Relu) (Nair and Hinton 2010) as the activation function. We tuned CFIN's parameters using 5-fold cross validation (CV). All the features are normalized before fed into CFIN. We test CFIN's performance on both KDDCUP dataset and XuetangX dataset. For KDDCUP dataset, the history period and prediction period are set to 30 days and 10 days by the competition organizer, respectively. We do not use the attention mechanism of CFIN on this data, as there is no context information provided in the competition. For XuetangX dataset, the history period is set to 35 days, prediction period is set to 10 days, i.e., $D_h = 35, D_p = 10$.

Comparison Methods. We conduct the comparison experiments for following methods:

- **LR**: logistic regression model.
- **SVM**: The support vector machine with linear kernel.
- **RF**: Random Forest model.
- **GBDT**: Gradient Boosting Decision Tree.
- **DNN**: 3-layer deep neural network.
- **CFIN**: The CFIN model.

[5] http://tensorflow.org

Table 5: Overall Results on KDDCUP dataset and IPM courses of XuetangX dataset.

Methods	KDDCUP AUC	KDDCUP F1	XuetangX AUC	XuetangX F1
LRC	86.78	90.86	82.23	89.35
SVM	88.56	91.65	82.86	89.78
RF	88.82	91.73	83.11	89.96
DNN	88.94	91.81	85.64	90.40
GBDT	89.12	91.88	85.18	90.48
CFIN	90.07	92.27	86.50	90.95
CFIN-en	90.93	92.87	86.75	90.99

Table 6: Contribution analyses for different kinds of engagements on KDDCUP dataset and IPM courses of XuetangX dataset.

Features	KDDCUP AUC	KDDCUP F1	XuetangX AUC	XuetangX F1
All	90.07	92.27	86.50	90.95
- Video activity	87.40	91.61	84.40	90.32
- Forum activity	88.61	91.93	85.13	90.41
- Assignment activity	86.68	91.39	84.83	90.34
- (Video + Forum)	84.70	91.4	84.42	90.31
- (Video + Assignment)	84.32	91.17	83.14	89.63
- (Forum + Assignment)	85.07	91.49	84.72	90.29

- **CFIN-en**: The assembled CFIN using the strategy proposed in Model Ensemble.

For baseline models (**LR, SVM, RF, GBDT, DNN**) above, we use all the features (including learning activity $\mathbf{X}$ and context information $\mathbf{Z}$) as input. When training the models, we tune the parameters using the average AUC on 5-fold cross validation (CV) with the grid search algorithm, and use the best group of parameters in all experiments.

Evaluation Measures. We evaluate the classification performance by using the Area Under the ROC Curve (AUC) and F1 Score (F1).

Prediction performance

Table 5 presents the results on KDDCUP dataset and IPM courses of XuetangX dataset for all comparison methods. Overall, **CFIN-en** gets the best performance on both two datasets, and its AUC score on KDDCUP dataset achieves 90.93% which is comparable to the winning team of KDDCUP 2015[6]. **CFIN** also beats the other baseline methods. Compared to LR and SVM, **CFIN** achieves 1.51 – 3.29% and 3.64 – 4.27% AUC score improvements on KDDCUP dataset and XuetangX dataset, respectively. Moreover, compared to the ensemble methods (i.e. RF and GBDT) and DNN, CFIN also shows better performance.

Feature Importance

In order to identify the importance of different kinds of engagement activities in this task, we conduct feature ablation experiments for three major activity features, i.e., video activity, assignment activity and forum activity, on two datasets. Specifically, we first input all the features to the CFIN, then remove every type of activity features one by one to watch the variety of performance. The results are shown in Table 6. We can observe that all the three kinds of engagements are useful in this task. For KDDCUP dataset, assignment plays the most important role. While for XuetangX dataset, video is more useful than the forum and assignment.

We also perform a fine-grained analysis for features' effects on different users. Specifically, we feed a set of typical features into CFIN, and compute their average attention weights for each cluster. The results are shown in Table 7. We can observe that the distributions of attention weights on the five clusters are quite different. The most significant difference appears in CAR (correct answer ratio): Its attention weight on cluster 5 (hard workers) is much higher than those on other clusters, which indicates that correct answer ratio is most important in predicting dropout for hard workers. While for users with more forum activities (cluster 2), answering questions in forum seems to be the key factor, as the corresponding attention weight of the number of answering questions is the highest. Another interesting thing is about the users with high dropout rates (cluster 1, 3 and 4). They get much higher attention wights on the number of stopping video and watching video compared to cluster 2 and cluster 5. Though video activities do not matter for active learners, they play an important role for students with poor engagements.

Table 7: Average attention weights of different clusters. C1-C5 — Cluster 1 to 5; CAR — average correct answer ratio.

Category	Type	C1	C2	C3	C4	C5
video	#watch	0.078	0.060	0.079	0.074	0.072
	#stop	0.090	0.055	0.092	0.092	0.053
	#jump	0.114	0.133	0.099	0.120	0.125
forum	#question	0.136	0.127	0.138	0.139	0.129
	#answer	0.142	0.173	0.142	0.146	0.131
assignment	CAR	0.036	0.071	0.049	0.049	0.122
	#reset	0.159	0.157	0.159	0.125	0.136
session	seconds	0.146	0.147	0.138	0.159	0.151
	count	0.098	0.075	0.103	0.097	0.081

Table 8: Results on SPM courses

Methods	LRC	SVM	RF	GBDT	CFIN	CFIN-en
AUC	69.76	71.23	72.34	72.21	73.47	74.11

Discussion for SPM Courses

Besides experiments on IPM courses, we also conduct the same experiments on SPM courses of XuetangX dataset. As shown in table 8, all the prediction results on SPM courses are rather lower than the results on IPM courses (table 5). This is because SPM courses give more freedom to students, and do not require students to learn within a specified time. Therefore, there is more uncertainty for dropout prediction

[6]https://biendata.com/competition/kddcup2015/rank/

(a) Strategy 1. (b) Strategy 2. (c) Strategy 3.

Figure 6: The snapshots of three intervention strategies.

Table 9: Results of A/B test. WVT — average watching video time (s); ASN — average assignment submitted num; CAR — average correct answer ratio.

Activity	No intervention	Strategy 1	Strategy 2	Strategy 3
WVT	4736.04	4774.59	5969.47	3402.96
ASN	4.59	9.34*	2.95	11.19**
CAR	0.29	0.34	0.22	0.40

on SPM courses.

Online Intervention

We deployed our framework on the *XiaoMu* system, an intelligent teaching assistant system on XuetangX, for facilitating user retention. For each course, we first provide online dropout prediction for all enrolled users. Then if a user's dropout probability is greater than a threshold, *XiaoMu* would encourage her to learn on the courses by sending him a reminder message. Specifically, we consider three different intervention strategies:

- **Strategy 1.** Send user a message of *"Based on our study, the probability of you obtaining a certificate can be increased by about 3% for every hour of video watching." when user come to a course.*
- **Strategy 2.** Send user the same message as that of strategy 1 when user watching video.
- **Strategy 3.** Send user a message of *"You spent xxx minutes learning and completed xxx homework questions in last week, keep it up!"* when user come to a course.

Figure 6 shows the snapshots of three strategies. We run an A/B test on four courses (i.e. Financial Analysis and Decision Making, Introduction to Psychology, C++ Programming and Java Programming) to examine the effectiveness of different strategies. The users in these courses are split into four groups, including three treatment groups corresponding to three intervention strategies and one control group. We collect two weeks of data and examine the video activities and assignment activities of different group of users. Table 9 shows the results. We also report t-test results between the control group and each treatment group: * means $p < 0.1$, ** means $p < 0.05$. We can find strategy 1 and strategy 3 can significantly improve users' enthusiasm for doing homework. While strategy 2 seems more effective in encouraging users to watch videos.

Conclusion

In this paper, we conduct a systematical study for the dropout problem in MOOCs. We first conduct statistical analyses to identify factors which may influence users' dropout. This includes correlation with other courses, influence from friends. These observations are useful for teachers and MOOCs' designers to better navigate and design courses and platforms in the future. Based on these analyses, we propose a context-aware feature interaction network (CFIN) to predict users' dropout. Our method gets the best performance on both KDDCUP dataset and XuetangX dataset, and the model have also been deployed online on XuetangX platform to improve students retention.

Acknowledgements

Another author of this paper is Shuhuai Zhang, from PBC School of Finance, Tsinghua University.

References

Anderson, A.; Huttenlocher, D.; Kleinberg, J.; and Leskovec, J. 2014. Engaging with massive online courses. In *Proceedings of the 23rd International Conference on World Wide Web*, 687–698.

Balakrishnan, G., and Coetzee, D. 2013. Predicting student retention in massive open online courses using hidden markov models. *Electrical Engineering and Computer Sciences University of California at Berkeley*.

Bayer, J.; Bydzovská, H.; Géryk, J.; Obsivac, T.; and Popelinsky, L. 2012. Predicting drop-out from social behaviour of students. *International Educational Data Mining Society*.

Chaturvedi, S.; Goldwasser, D.; and Daumé III, H. 2014. Predicting instructor's intervention in mooc forums. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 1501–1511.

Chen, T., and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Dalipi, F.; Imran, A. S.; and Kastrati, Z. 2018. Mooc dropout prediction using machine learning techniques: Review and research challenges. In *Global Engineering Education Conference (EDUCON), 2018 IEEE*, 1007–1014. IEEE.

Fei, M., and Yeung, D.-Y. 2015. Temporal Models for Predicting Student Dropout in Massive Open Online Courses. *2015 IEEE International Conference on Data Mining Workshop (ICDMW)* 256–263.

Guo, P. J.; Kim, J.; and Rubin, R. 2014. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, 41–50.

Halawa, S.; Greene, D.; and Mitchell, J. 2014. Dropout prediction in moocs using learner activity features. *Experiences and best practices in and around MOOCs* 3–12.

He, J.; Bailey, J.; Rubinstein, B. I. P.; and Zhang, R. 2015. Identifying at-risk students in massive open online courses. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 1749–1755.

He, J.; Rubinstein, B. I. P.; Bailey, J.; Zhang, R.; Milligan, S.; and Chan, J. 2016. Moocs meet measurement theory: A topic-modelling approach. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 1195–1201.

Kellogg, S. 2013. Online learning: How to make a mooc. *Nature* 369–371.

Kim, J.; Guo, P. J.; Seaton, D. T.; Mitros, P.; Gajos, K. Z.; and Miller, R. C. 2014. Understanding in-video dropouts and interaction peaks inonline lecture videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, 31–40.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kizilcec, R. F.; Piech, C.; and Schneider, E. 2013. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, 170–179.

Kloft, M.; Stiehler, F.; Zheng, Z.; and Pinkwart, N. 2014. Predicting MOOC Dropout over Weeks Using Machine Learning Methods. 60–65.

Nagrecha, S.; Dillon, J. Z.; and Chawla, N. V. 2017. Mooc dropout prediction: Lessons learned from making pipelines interpretable. In *Proceedings of the 26th International Conference on World Wide Web Companion*, 351–359.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Onah, D. F.; Sinclair, J.; and Boyatt, R. 2014. Dropout rates of massive open online courses: behavioural patterns. *EDULEARN'14* 5825–5834.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 701–710.

Qiu, J.; Tang, J.; Liu, T. X.; Gong, J.; Zhang, C.; Zhang, Q.; and Xue, Y. 2016. Modeling and predicting learning behavior in moocs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 93–102.

Ramesh, A.; Goldwasser, D.; Huang, B.; Daumé, III, H.; and Getoor, L. 2014. Learning latent engagement patterns of students in online courses. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1272–1278.

Ramesh, A.; Kumar, S. H.; Foulds, J.; and Getoor, L. 2015. Weakly supervised models of aspect-sentiment for online course discussion forums. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 74–83.

Reich, J. 2015. Rebooting mooc research. *Science* 34–35.

Rousseeuw, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20:53–65.

Seaton, D. T.; Bergner, Y.; Chuang, I.; Mitros, P.; and Pritchard, D. E. 2014. Who does what in a massive open online course? *Communications of the Acm* 58–65.

Shah, D. 2018. A product at every price: A review of mooc stats and trends in 2017. *Class Central*.

Wang, W.; Yu, H.; and Miao, C. 2017. Deep model for dropout prediction in moocs. In *Proceedings of the 2nd International Conference on Crowd Science and Engineering*, 26–32. ACM.

Whitehill, J.; Williams, J.; Lopez, G.; Coleman, C.; and Reich, J. 2015. Beyond prediction: First steps toward automatic intervention in mooc student stopout.

Wolpert, D. H. 1992. Stacked generalization. *Neural networks* 5(2):241–259.

Xing, W.; Chen, X.; Stein, J.; and Marcinkowski, M. 2016. Temporal predication of dropouts in moocs: Reaching the low hanging fruit through stacking generalization. *Computers in Human Behavior* 119–129.

Yang, D.; Sinha, T.; Adamson, D.; and Rosé, C. P. 2013. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop*, 14.

Zheng, S.; Rosson, M. B.; Shih, P. C.; and Carroll, J. M. 2015. Understanding student motivation, behaviors and perceptions in moocs. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work*, 1882–1895.

Zhenghao, C.; Alcorn, B.; Christensen, G.; Eriksson, N.; Koller, D.; and Emanuel, E. 2015. Whos benefiting from moocs, and why. *Harvard Business Review* 25.