A Nonlinear State Space Model for Identifying At-Risk Students in Open Online Courses

Feng Wang and Li Chen
Department of Computer Science
Hong Kong Baptist University, Hong Kong
{fwang, lichen}@comp.hkbu.edu.hk

ABSTRACT

How to identify at-risk students in open online courses has received increasing attention, since the dropout rate is unexpectedly high. Most prior studies have focused on using machine learning techniques to predict student dropout based on features extracted from students' learning activity logs. However, little work has viewed the dropout prediction problem as a sequence classification problem in the consideration that the dropout probability of a student at the current time step can be likely dependent on her/his engagement at the previous time step. Therefore, in this paper, we propose a nonlinear state space model to solve this problem. We show how students' latent states at different time steps can be learned via this model, and demonstrate its outperforming prediction accuracy relative to related methods through experiment.

Keywords

At-risk students; Dropout prediction; Open online courses, Nonlinear state space model

1. INTRODUCTION

With the advent of open online courses, such as MOOC websites Edx, Coursera, Khan Academy, high quality education can easily be accessed by students at low cost. However, although many thousands of participants have enrolled on the online courses, their dropout rate is extremely higher than expected. As reported in [8], the average dropout rate of current MOOCs is approximately 75%.

Identifying at-risk students by predicting their dropout probability thus becomes timely important, given that early prediction can help instructors provide proper support to those students to retain their learning interests. To address this issue, some researchers focused on extract features from students' learning activities (such as watching videos, working on assignments, and posting in or viewing discussion forums) for building machine learning models (like support vector machine (SVM) [9] and logistic regression (LG) [14]). However, they rarely considered that students' learning activities across different time steps (e.g., weeks) might be interrelated and take different weights in making the prediction. For instance, recent activities could be more important to reflect students' engagement degree. If a student actively engages with a course in the current week, it is more likely that s/he will continue to engage with this course in the coming week. Otherwise, if s/he becomes inactive, it may infer that her/his interest in the course is decreased. Recently, though some approaches, such as the one based on Hidden Markov Model (HMM) [2] and that based on Recurrent Neural Network (RNN) [12], have been proposed to model students' states over time, they still suffer from some issues: 1) the estimation of next state depends only on the current state; 2) the estimated states are deterministic that would lead to error propagation in the estimation procedure; 3) the parameters of their models are time-invariant.

In our work, we focus on predicting whether a student will have activities in the coming week. We particularly formulate this issue as *sequential classification* problem, and develop *Nonlinear State Space Model* (NSSM) [1] to solve it. Essentially, NSSM has several advantages. Firstly, it can be used to discover a student's latent state (i.e., *engagement pattern*) to characterize the student's intention to perform certain activities. The student's dropout probability is then computed based on the state estimated for that time. Secondly, relative to HMM and RNN, NSSM takes into account all of the current and previous states to estimate next state. It can also accommodate uncertainty given that the state in NSSM is a set of random variables with *multivariate Gaussian distribution*. Thirdly, the parameters in NSSM are time varying (i.e., being different at different time steps), which makes it more flexible to model students' dynamics.

In short, this paper has two main contributions: 1) we implement Nonlinear State Space Model (NSSM) to address the dropout prediction problem, which particularly models students' latent states varying over time; 2) we conduct experiment to compare our method with related ones including logistic regression (LG), simultaneously smoothed logistic regression (LR-SIM), and RNN with long short-term memory cell (LSTM). It shows that our method is more accurate in identifying at-risk students who tend to drop out.

In the remainder, we first describe related work in Section 2, and then present our methodology in Section 3. In Section 4,

we give experimental results. In Section 5, we conclude our work and indicate its future directions.

2. RELATED WORK

High dropout rate that popularly exists in current MOOCs has driven some researchers to investigate the issue of identifying at-risk students who are likely to quit. They have considered different features to build the prediction model, such as those extracted from clickstream data (e.g., watching a lecture video, posting to discuss forums, submitting an assignment) [2, 5, 6, 9, 14], quiz performance [5, 6, 14], centrality of students in discussion forums [15], and sentiments of discussion forum posts [4].

As for prediction model, some studies have applied support vector machines (SVM) [9], logistic regression (LG) [14], survival analysis techniques like Cox proportional hazard model [15], and probabilistic soft logic (PSL) [13]. However, their common limitation is that they assume a student's dropout probabilities at different time steps are independent, which limits the approach's applicability in practice as usually a student's state at one time can be influenced by her/his previous state.

Alternatively, [6] extended logistic regression model to smooth the dropout probabilities across weeks with the aim to minimize the difference of successing predicted probabilities between weeks. [2] used Hidden Markov Model (HMM) to model student's actions over time, which encodes their behaviour features into a set of mutually exclusive discrete states. [12] adopted Recurrent Neural Network (RNN) model with long short-term memory (LSTM) cells, which is able to encode features into continuous states. However, though RNN may be advantageous against HMM, it inherently suffers from error propagation phenomenon because the estimation of current state depends only on the estimated previous state.

In comparison, in our model, the uncertainty of estimated states is considered by representing the state as random variables drawing from a multivariate Gaussian distribution. What's more, we adopt extended Kalman filter and smoother for state estimation so as to take into account all observed activities in sequence, which makes it different from, and potentially more effective than, HMM and RNN where only states at two consecutive time steps are related.

3. OUR METHODOLOGY

3.1 Problem Statement

As mentioned above, our goal is to estimate the probability that a student stops engaging with a course in the coming week, given her/his learning activities up to the current time step.

The temporal prediction of dropout probability requires us to assemble some features [1] for expressing time-varying behavior of students. Therefore, we extract 28 typical features for each week t, denoted as N dimensional vector $\mathbf{x}_{i,t} \in \mathbb{R}^N$, by considering the seven types of activity [2]. The summarization of these temporal features is listed in Table 1.

Table 1: List of features derived from each student's learning activities by the week t

Features	Description
x_1	The average number of activities per week by the week t.
x_2	The total number of activities in week t.
x_3	The average number of sessions per week by the week t. [3]
x_4	The total number of sessions in week t.
x_5	The average number of active days per week by the week t. [4]
x_6	The total number of active days in week t.
x_7	The average time consumption per week by the week t.
x_8	The total time consumption in week t.
x_9 - x_{15}	The average number of 7 different types of activity per week by the week t.
x_{16} - x_{22}	The total number of 7 different types of activity in week t.
$x_{23} - x_{25}$	The average number of videos watched, wiki viewed and problem attempted per session by the week t respectively.
$x_{26} - x_{28}$	The average number of videos watched, wiki viewed and problem attempted per session in week t respectively.

In consequence, we obtain a sequence $(\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,n_i})$ for each student i across n_i weeks, as well as the corresponding sequence of dropout labels $(y_{i,1}, y_{i,2} \ldots, y_{i,n_i})$. Here n_i represents the number of weeks during which student i has engaged with the course. Formally, for current week t, if there are activities associated to student i in the coming week, her/his dropout label in the week t is assigned $y_{i,t} = 0$, otherwise $y_{i,t} = 1$. We can then treat the dropout prediction task as a *sequential classification* problem, for which the student's latent states evolving over time are not observable directly. As illustrated in Figure 1, as the course progresses, given the student i's features $\mathbf{x}_{i,t}$ for the current week t, and his/her previous state $\mathbf{s}_{i,t-1}$, we want to estimate the student's current state $\mathbf{s}_{i,t}$ and whether s/he will continue engaging with the course in the coming week $y_{i,t}$.

3.2 Nonlinear State Space Model (NSSM)

Specifically, we employ a nonlinear state space model (NSSM) with continuous value states to summarize all the information about a student's past behavior. Formally, let the vector $\mathbf{s}_{i,t} \in \mathbb{R}^K$ ($K \ll N$) be the latent state of student i in the t-th week, which depends on the observed explanatory features $\mathbf{x}_{i,t}$ and her/his previous state $\mathbf{s}_{i,t-1}$, as follows:

$$\mathbf{s}_{i,t} = \mathbf{F}\mathbf{s}_{i,t-1} + \mathbf{G}\mathbf{x}_{i,t} + \mathbf{w}_{i,t} \quad (1)$$

in which the matrix $\mathbf{F} \in \mathbb{R}^{K \times K}$ transforms the previous state into the current state, the matrix $\mathbf{G} \in \mathbb{R}^{K \times N}$ transforms the observed features to reflect the current state, and

[1] Prior to model training, these features are normalized to have mean 0 and variance 1, and the normalization parameters (mean, standard deviation) are used for normalizing the testing set.

[2] The seven types of activity consist of watching lecture videos, working on course's problems, accessing course's modules, accessing course's wiki, posting or viewing course's forum, navigating through courses, and closing course page.

[3] The minimal elapsed time between two separate sessions is set as 60 minutes.

[4] The day that has at leas one activity is treated as an active day.

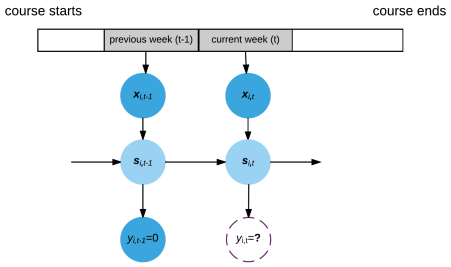

Figure 1: The illustration of MOOCs dropout prediction problem and the graphical state space model. The dark blue signifies an observed variable and the light blue signifies a latent variable.

$\mathbf{w}_{i,t}$ represents a diffusion variable which follows a multivariate Gaussian with mean $\mathbf{0}$ and covariance $\mathbf{Q}_{i,t}$ (i.e., $\mathbf{w}_{i,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{i,t})$). Note that the dimension of the state vector K is usually smaller than the dimension of feature vector N. This hyperparameter K controls the complexity of the model, and requires manual tuning to determine its optimal value.

In our work, we aim to infer the dropout probability $\pi_{i,t}$ for student i in week t, which can be represented as logistic regression

$$\pi_{i,t} = \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t} + \boldsymbol{\beta}_t^T \mathbf{x}_{i,t}) \quad (2)$$

$$= \frac{1}{1 + \exp(-\mathbf{h}_t^T \mathbf{s}_{i,t} - \boldsymbol{\beta}_t^T \mathbf{x}_{i,t})} \quad (3)$$

where $\mathbf{h}_t \in \mathbb{R}^{K \times 1}$ and $\boldsymbol{\beta}_t \in \mathbb{R}^{N \times 1}$ are two vectors of coefficients for current state variable $\mathbf{s}_{i,t}$ and input feature $\mathbf{x}_{i,t}$ respectively. In this model, the non-stationary of student dynamic is captured by time-evolving state variable $\mathbf{s}_{i,t}$, and time-varying parameters $\mathbf{h}_t$ and $\boldsymbol{\beta}_t$.

3.3 Expectation Maximization

With the nonlinear state space model described in Eqn. 1 and Eqn. 2, we design an Expectation-Maximization (EM) algorithm (see Algorithm 1) that iterates between state estimation (E-step) and parameter estimation (M-step) [11]. The E-step makes use of extended Kalman *filter* and *smoother* to estimate states, and the M-step re-estimates the parameters by maximizing the likelihood of all observed data, in which the state variables of student are replaced by their posteriori values from the extended Kalman smoother.

3.3.1 Expectation Step

In the expectation step, the expected mean of student state $\mathbf{s}_{i,t}$ and its covariance $\mathbf{P}_{i,t}$ are obtained using the extended Kalman filter and smoother. Specifically, given student i's entire $t-1$ weeks' observation sequence $D_i^{(t-1)} = \{(\mathbf{x}_{i,1}, y_{i,1}), (\mathbf{x}_{i,2}, y_{i,2}), \ldots, (\mathbf{x}_{i,t-1}, y_{i,t-1})\}$, the posterior mean and covariance of student state $\mathbf{s}_{i,t-1}$ are supposed be represented by $E(\mathbf{s}_{i,t-1}|D_i^{(t-1)}) = \mathbf{s}_{i,t-1}^{(t-1)}$ and $Cov(\mathbf{s}_{i,t-1}|D_{i,t-1}) = \mathbf{P}_{i,t-1}^{(t-1)}$ respectively. The predicted student state $\mathbf{s}_{i,t}$ and its covariance $\mathbf{P}_{i,t}^{(t-1)}$ for $t = 1, 2, \ldots, n_i - 1, n_i$ can then be defined as:

Algorithm 1 EM algorithm for estimating latent student state and model parameters.

1: Initialize each student's starting state $\mathbf{s}_{i,0}$ and model parameters $\Phi = \{\mathbf{F}, \mathbf{G}, \mathbf{h}_t, \boldsymbol{\beta}_t\}$
2: **repeat**
3: **procedure E-step:**
4: **Extended Kalman filter:** For $t = 1, 2, \ldots, n_i - 1, n_i$, correct the student state $\mathbf{s}_{i,t}$ and its covariance $\mathbf{P}_{i,t}$ by using Eqn. 10 and Eqn. 11 respectively.
5: **Extended Kalman smoother:** For $t = n_i, n_i - 1, \ldots, 2, 1$, smooth the predicted student state $\mathbf{s}_{i,t}^{(t)}$ and covariance $\mathbf{P}_{i,t}^{(t)}$ by using Eqn. 13 and Eqn. 14 respectively.
6: **end procedure**
7: **procedure M-step:**
8: Update parameters of the model Φ via equations from Eqn. 17 to Eqn. 20.
9: **end procedure**
10: **until** converged

$$\mathbf{s}_{i,t}^{(t-1)} = \mathbf{F}\mathbf{s}_{i,t-1}^{(t-1)} + \mathbf{G}\mathbf{x}_{i,t} \quad (4)$$

$$\mathbf{P}_{i,t}^{(t-1)} = \mathbf{F}\mathbf{P}_{i,t-1}^{(t-1)}\mathbf{F}^T + \mathbf{Q}_{i,t} \quad (5)$$

By following the extended Kalman filtering, the nonlinear function $\sigma(\cdot)$ can be approximated by its Taylor series expansion as follows:

$$\begin{aligned} \pi_{i,t} &= \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t} + \boldsymbol{\beta}_t^T \mathbf{x}_{i,t}) \\ &\approx \sigma(\mathbf{h}_t \mathbf{s}_{i,t}^{(t-1)} + \boldsymbol{\beta}_t^T \mathbf{x}_{i,t}) + \mathbf{A}_{i,t}^T (\mathbf{s}_{i,t} - \mathbf{s}_{i,t-1}^{(t-1)}) \end{aligned} \quad (6)$$

where

$$\begin{aligned} \mathbf{A}_{i,t} &\triangleq \frac{\partial \sigma(\mathbf{h}_{i,t}^T \mathbf{s}_{i,t} + \boldsymbol{\beta}_t^T \mathbf{x}_{i,t})}{\partial \mathbf{s}_{i,t}} \\ &= \sigma\left(\mathbf{h}_{i,t}^T \mathbf{s}_{i,t}^{(t-1)} + \boldsymbol{\beta}_t^T \mathbf{x}_{i,t}\right) \\ &\quad \left(1 - \sigma(\mathbf{h}_{i,t}^T \mathbf{s}_{i,t}^{(t-1)} + \boldsymbol{\beta}_t^T \mathbf{x}_{i,t})\right) \mathbf{h}_{i,t} \end{aligned} \quad (7)$$

The one-step ahead prediction $\pi_{i,t}^{(t-1)}$ for the dropout probability is computed as:

$$\pi_{i,t}^{(t-1)} = \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t}^{(t-1)} + \boldsymbol{\beta}_t^T \mathbf{x}_{i,t}) \quad (8)$$

For the sake of simplicity, we set the state noise covariance as $\mathbf{Q}_{i,t} = q_{i,t}\mathbf{I}$, where the state noise variance $q_{i,t}$ is computed via:

$$q_{i,t} = max\{\mu_{i,t}^{(t)} - \mu_{i,t}^{(t-1)}, 0\} \quad (9)$$

in which $\mu_{i,t}^{(\cdot)} = \pi_{i,t}^{(\cdot)}(1 - \pi_{i,t}^{(\cdot)})$. After receiving a new observation $(\mathbf{x}_{i,t}, y_{i,t})$, the predicted state $\mathbf{s}_{i,t}^{(t-1)}$ in Eqn. 4 and covariance $\mathbf{P}_{i,t}^{(t-1)}$ in Eqn. 5 will be updated as:

$$\mathbf{s}_{i,t}^{(t)} = \mathbf{s}_{i,t}^{(t-1)} + \mathbf{K}_{i,t}\left(y_{i,t} - \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t}^{(t-1)} + \boldsymbol{\beta}_t^T \mathbf{x}_{i,t})\right) \quad (10)$$

$$\mathbf{P}_{i,t}^{(t)} = (\mathbf{I} - \mathbf{K}_{i,t}\mathbf{A}_{i,t})\mathbf{P}_{i,t}^{(t-1)} \quad (11)$$

in which $\mathbf{K}_{i,t}$ is the Kalman gain computed according to [3]:

$$\mathbf{K}_{i,t} = \mathbf{P}_{i,t}^{(t-1)}\mathbf{A}_{i,t}^T\left(\mathbf{A}_{i,t}\mathbf{P}_{i,t}^{(t-1)}\mathbf{A}_{i,t}^T + \mathbf{Q}_{i,t}\right)^{-1} \quad (12)$$

It is worth noting that the predicted state $\mathbf{s}_{i,t}^{(t)}$ and covariance $\mathbf{P}_{i,t}^{(t)}$ in Kalman filter are estimated based on the observation $D_i^{(t)}$ up to week t. We take advantage of extended

Kalman smoother to smooth the estimated states by considering the entire sequence of the student's observations $D_i^{(n_i)}$. The smoothed states could hence be more accurate than the filtered ones. Specifically, the student state $\mathbf{s}_{i,t-1}^{(n_i)}$ and covariance $\mathbf{P}_{i,t-1}^{(n_i)}$ for $t = n_i, n_i - 1, \ldots, 1$ are recursively smoothed as:

$$\mathbf{s}_{i,t-1}^{(n_i)} = \mathbf{s}_{i,t-1}^{(t-1)} + \mathbf{J}_{i,t-1}\left(\mathbf{s}_{i,t}^{(n_i)} - \mathbf{F}\mathbf{s}_{i,t-1}^{(t-1)} - \mathbf{G}\mathbf{x}_{i,t-1}\right) \quad (13)$$

$$\mathbf{P}_{i,t-1}^{(n_i)} = \mathbf{P}_{i,t-1}^{(t-1)} + \mathbf{J}_{i,t-1}\left(\mathbf{P}_{i,t}^{(n_i)} - \mathbf{P}_{i,t}^{(t-1)}\right)\mathbf{J}_{i,t-1}^T \quad (14)$$

where $\mathbf{J}_{i,t-1}$ is the smoothing gain defined as:

$$\mathbf{J}_{i,t-1} = \mathbf{P}_{i,t-1}^{(t-1)}\mathbf{F}^T\left(\mathbf{P}_{i,t}^{(t-1)}\right)^{-1} \quad (15)$$

Note that the initial values $\mathbf{s}_{i,n_i}^{(n_i)}$ and $\mathbf{P}_{i,n_i}^{(n_i)}$ for the smoother are the final estimates of the filter.

3.3.2 *Maximization Step*

At the maximization step, given the observed data D of N students, the likelihood is defined as

$$\begin{aligned}\mathcal{L}(D|\Phi) = &\sum_{i=1}^{N}\sum_{t=1}^{n_i} y_{i,t}\log(\sigma(\mathbf{h}_{i,t}^T\mathbf{s}_{i,t}^{(n_i)} + \boldsymbol{\beta}_t^T\mathbf{x}_{i,t})) \quad (16)\\ &+ (1-y_{i,t})\log(1-\sigma(\mathbf{h}_{i,t}^T\mathbf{s}_{i,t}^{(n_i)} + \boldsymbol{\beta}_t^T\mathbf{x}_{i,t}))\\ &- \frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{n_i}(\mathbf{s}_{i,t}^{(n_i)} - \mathbf{F}\mathbf{s}_{i,t-1}^{(n_i)} - \mathbf{G}\mathbf{x}_{i,t})^T\mathbf{Q}_{i,t}^{-1}(\mathbf{s}_{i,t}^{(n_i)}\\ &- \mathbf{F}\mathbf{s}_{i,t-1}^{(n_i)} - \mathbf{G}\mathbf{x}_{i,t}) - \frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{n_i}\log|\mathbf{Q}_{i,t}|\end{aligned}$$

By using the posterior hidden state variables $\mathbf{s}_{i,t}^{(n_i)}$ from Kalman smoother, the optimal parameters $\Phi = \{\mathbf{G}, \mathbf{F}, \mathbf{h}_t, \boldsymbol{\beta}_t\}$ can be obtained by maximizing the likelihood defined in Eqn. 16. We then apply the gradient based method L-BFGS [10] to update model parameters by using the following derivation formulas respectively:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{F}} = -\sum_{i=1}^{N}\sum_{t=1}^{n_i}\left(\mathbf{s}_{i,t}^{(n_i)} - \mathbf{F}\mathbf{s}_{i,t-1}^{(n_i)} - \mathbf{G}\mathbf{x}_{i,t}\right)\mathbf{Q}_{i,t}^{-1}\mathbf{s}_{i,t-1}^{(n_i)} \quad (17)$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{G}} = -\sum_{i=1}^{N}\sum_{t=1}^{n_i}\left(\mathbf{s}_{i,t}^{(n_i)} - \mathbf{F}\mathbf{s}_{i,t-1}^{(n_i)} - \mathbf{G}\mathbf{x}_{i,t}\right)\mathbf{Q}_{i,t}^{-1}\mathbf{x}_{i,t} \quad (18)$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{h}_t} = \sum_{i=1}^{N}\sum_{t=1}^{n_i}\left(y_{i,t} - \sigma(\mathbf{h}_t^T\mathbf{s}_{i,t}^{(n_i)} + \boldsymbol{\beta}_t^T\mathbf{x}_{i,t})\right)\mathbf{s}_{i,t}^{(n_i)} \quad (19)$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}_t} = \sum_{i=1}^{N}\sum_{t=1}^{n_i}\left(y_{i,t} - \sigma(\mathbf{h}_t^T\mathbf{s}_{i,t}^{(n_i)} + \boldsymbol{\beta}_t^T\mathbf{x}_{i,t})\right)\mathbf{x}_{i,t} \quad (20)$$

Initialization of the EM Algorithm: The initial value of parameters Φ should be chosen with care, otherwise the EM algorithm may not converge. In our experiment, the matrix $\mathbf{G}$ is initially set as the transform matrix resulted from principle component analysis (PCA) algorithm [7], and the matrix $\mathbf{F}$ is assigned to be an identity matrix.

4. EXPERIMENT

In order to evaluate the performance of our proposed model, we conducted an experiment on a real-life dataset.

4.1 Dataset

We use a data set collected from xuetangX [5], one of the largest MOOC platforms in China. This dataset was released for KDD CUP 2015 [6]. The dataset, as shown in Table 2, includes 79,186 students each of whom enrolled on at least one course among the whole set of 39 courses. Each enrollment is associated with a log of the student's activities including watching lecture videos, working on course's problems, accessing course's modules, and so on. Totally, there are 8,157,277 activity logs and the longest lifetime of enrollment is 5 weeks.

Table 2: Statistics of xuetangX dataset for the experiment

Item	Statistical description
# courses	39
# students	79,186
# enrollments	120,542
# activity logs	8,157,277
# longest lifetime of enrollment	5 weeks

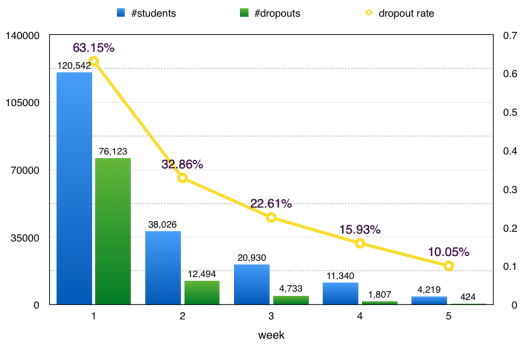

Figure 2: The number of students, number of dropouts, and the dropout rate in different weeks.

As shown in Figure 2, we observe that 76,123 students dropped out in the first week. Another observation is that the longer the student has engaged with the course, the less likely s/he quit the course. For example, the dropout rate of students who have engaged with the courses for 5 weeks is 10.05% vs. 63.15% for 1 week.

4.2 Evaluation Metrics

Due to the class imbalance phenomenon, we use Area Under the Receiver Operating Characteristics Curve (AUC) as the evaluation metric, as it is invariant to imbalance. Concretely, AUC measures how likely a classifier can correctly discriminate between positive and negative samples. An AUC of 1 indicates perfect discrimination whereas 0.5 corresponds to a classifier that guesses randomly.

[5]http://www.xuetangx.com
[6]http://www.kddcup2015.com

4.3 Compared Methods

We compared our model with related methods:

- Logistic Regression (LG) [14]: In this method, a logistic regression classifier is trained to make dropout prediction for each week. Specifically, for a student i in week t, his/her dropout probability is computed as the logistic function of the weighted sum of input features $\mathbf{x}_{i,t}$:

$$p(y_{i,t}|\mathbf{x}_{i,t}, \mathbf{w}_t) = \frac{1}{1+\exp(-y_{i,t}\mathbf{w}_t^T\mathbf{x}_{i,t})} \quad (21)$$

  where $\mathbf{w}_t = [w_{t1}, w_{t2}, \ldots, w_{tN}]^T$ is the weight vector to be learned. The objective function for week t is

$$\mathcal{L}(\mathbf{w}_t) = \sum_{i\in N_t} \log(1+\exp(-y_{i,t}\mathbf{w}_t^T\mathbf{x}_{i,t})) + \frac{\lambda_1}{2}||\mathbf{w}_t||^2 \quad (22)$$

  where N_t is the set of students who engage with the course in week t and $\lambda_1 > 0$ is the regularization parameter for $\mathbf{w}_t$.

- Simultaneously Smoothed Logistic Regression (LR-SIM) [6]: It extends the logistic regression by smoothing the predicted dropout probabilities across consecutive weeks. In this model, a regularization term is added into the objective function to minimize the difference of the predicted probabilities between two adjacent weeks, such as $\mathbf{w}_t^T\mathbf{x}_{i,t}$ and $\mathbf{w}_{t-1}^T\mathbf{x}_{i,t-1}$. A new feature space $\mathbf{x}'_{i,t}$ is introduced, which has $T \times N$ dimensions (T is the total number of weeks), with the t-th component having N features corresponding to the features in the original feature space $\mathbf{x}_{i,t}$ for week t, and other $T-1$ components corresponding to zeroes. Then, a single weight vector $\mathbf{w}$ is introduced, which also has $T \times N$ dimensions corresponding to $\mathbf{x}'_{i,t}$. The final objective function is defined as:

$$\mathcal{L}(\mathbf{w}) = \sum_{i\in N_t}\sum_{t=1}^{n_i} \log\left(1+\exp(-y_{i,t}\mathbf{w}^T\mathbf{x}'_{i,t})\right) + \frac{\lambda_1}{2}||\mathbf{w}||^2 + \lambda_2 \sum_{t=2}^{T}\sum_{i\in N_{t,t-1}} ||\mathbf{w}^T\mathbf{x}'_{i,t} - \mathbf{w}^T\mathbf{x}'_{i,t-1}||^2 \quad (23)$$

  where $N_{t,t-1}$ is the set of students who engage with the course in both weeks t and $t-1$, and $\lambda_2 > 0$ is the regularization parameter for the difference of the resulted dropout probabilities between two adjacent weeks.

- RNN with Long Short-Term Memory Cell (LSTM) [12]: It uses a recurrent neural network (RNN) model with long short-term memory (LSTM) architecture to train a sequence classifier model that produces temporal prediction. Similar to our proposed model, given the student's week-by-week features and dropout labels $\{(\mathbf{x}_{i,t}, y_{i,t}), 1 \le t \le n_i\}$, the LSTM model is applied to estimate the student state, which can then be used to predict the student's future actions.

Note that we did not compare with Hidden Markov Model (HMM) based method [2] because it can be treated as a special case of RNN by representing student state as discrete variable. For all the compared models, we used the same set of features as input (see Table 1).

4.4 Results and Discussion

The main hyperparameter to determine the NSSM model's performance is the dimensionality of student state K (see Eqn. 1). We compared the performance of NSSM in terms of AUC with varying dimension of latent state K, and observed that the optimal value of K in most cases is 12. Therefore, in our experiment, we set K as 12 to train the NSSM model.

4.4.1 Single Course

In this setting, we trained a separate model for each course. To get sufficient data for training, we only consider the popular courses that include more than 5,000 students. After filtering, 6 popular courses are used in this experiment. As students may enroll in a course at different time steps, we select 70% students who enrolled in the course in early period as the training data, and remaining 30% students as the testing data.

	LR	LR-SIM	LSTM	NSSM
Week 1	0.812	0.886	0.891	**0.900**
Week 2	0.819	0.876	0.887	**0.891**
Week 3	0.807	0.854	0.861	**0.870**
Week 4	0.768	0.778	0.786	**0.796**
Week 5	0.673	0.679	0.689	**0.702**

Table 3: Performance comparison of LR, LR-SIM, LSTM and NSSM in terms of average AUC on 6 popular courses.

Table 3 presents the average AUC scores across weeks by testing different models. The results indicate that the models that consider dependence between consecutive weeks, such as LR-SIM, LSTM and NSSM, achieve higher AUC score than the baseline LR model without this consideration. For example, for the first week, the AUC score of NSSM is 0.9, which is 10.8% improvement relative to that of LR model. Furthermore, we can see that the methods that model the student's states over time (i.e., LSTM and NSSM) achieve higher AUC than LR and LR-SIM in most cases. More notably, our proposed model NSSM performs consistently better than LSTM, suggesting that the student states estimated by NSSM is more predictive than those by LSTM. We can also observe that the accuracy during early weeks is higher than that of later weeks by most of models. This implies that the dropout prediction task may become harder with increasing lifetime of engagement, as there might be various hidden reasons that cause a student to quit the course.

4.4.2 Across Courses

In this setting, we are interested in evaluating whether the proposed model trained on some courses can serve other courses as well, for which we randomly select 70% courses for training and remaining 30% for testing. In this experiment, we use all of the student data from the training courses to train the model.

Table 4 shows the performance comparison. Same conclusions can be made as in the previous Section 4.4.1. Specifically, from this table, we can observe that our proposed model NSSM still outperforms the other models (e.g., LR, LR-SIM and LSTM) across different weeks. For example,

	LR	LR-SIM	LSTM	NSSM
Week 1	0.835	0.933	0.936	**0.936**
Week 2	0.911	0.915	0.915	**0.919**
Week 3	0.868	0.872	0.867	**0.871**
Week 4	0.782	0.784	0.785	**0.789**
Week 5	0.655	0.662	0.673	**0.686**

Table 4: Performance comparison of LR, LR-SIM, LSTM and NSSM in terms of AUC on new courses across weeks.

for the first week, the AUC score of NSSM is 0.686, which is 12% improvement relative to that of LR model. Furthermore, we can see that the improvement from NSSM with regard to LSTM is slight, and the relative improvement during later weeks is larger than that of early weeks (e.g., +5.1% during week 4 vs +4.4% during week 2). This observation implies that the NSSM has the potential to make better dropout predictions for students who have longer lifetime of engagement than LSTM. In addition, as these results are predictions made for students from new courses, we can conclude that our proposed model is capable of making better dropout prediction in new courses, in comparison with other models.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have focused on identifying at-risk students in online courses by making dropout prediction. We particularly take advantage of nonlinear state space model (NSSM) because it can discover a student's latent state to characterize the student's intention to perform certain activities. We conducted experiment on a real-world dataset, which demonstrates that our proposed model achieves higher prediction accuracy than related methods. We also showed that the NSSM model trained on data from some courses can make dropout prediction for students in new courses.

However, because the extended Kalman filter and smoother we used in this paper may not be an optimal parameter estimator, the difference between NSSM and LSTM is slight. Therefore, in the future, we will exploit other advanced algorithms (e.g., Unscented Kalman filter) to estimate the parameters in our nonlinear state space model. For the second future direction, as the experiment presented in this paper is limited to xuetangX dataset, we plan to evaluate our proposed model on datasets collected from other MOOC platforms, such as Edx and Coursera.

6. ACKNOWLEDGMENTS

This research work was supported by Hong Kong GRF ECS/HKBU211912 and partially supported by ITF ITS/271/14FX.

7. REFERENCES

[1] H. J. Andrew. *Stochastic Processes and Filtering Theory*. Academic Press, Inc., New York and London, 1970.

[2] G. Balakrishnan. Predicting student retention in massive open online courses using hidden markov models. Master's thesis, EECS Department, University of California, Berkeley, May 2013.

[3] M. Y. Byron, K. V. Shenoy, and M. Sahani. Derivation of extended kalman filtering and smoothing equations. Technical report, 2004.

[4] D. S. Chaplot, E. Rhim, and J. Kim. Predicting student attrition in moocs using sentiment analysis and neural networks. In *Proc. of the 2015 AIED Workshop on Intelligent Support for Learning in Groups*, pages 7–12, 2015.

[5] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in moocs using learner activity features. *Experiences and Best Practices in and around MOOCs*, 7:7–16, 2014.

[6] J. He, J. Bailey, B. I. P. Rubinstein, and R. Zhang. Identifying at-risk students in massive open online courses. In *Proc. of the 29th AAAI Conference on Artificial Intelligence*, pages 1749–1755, 2015.

[7] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

[8] K. Jordan. Mooc completion rates: The data. *Availabe at: http://www.katyjordan.com/MOOCproject.html. [Accessed: 04/02/2016]*, 2016.

[9] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting mooc dropout over weeks using machine learning methods. In *Proc. of the 2014 EMNLP Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 60–65, 2014.

[10] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

[11] W. Mader, Y. Linke, M. Mader, L. Sommerlade, J. Timmer, and B. Schelter. A numerically efficient implementation of the expectation maximization algorithm for state space models. *Applied Mathematics and Computation*, 241:222–232, 2014.

[12] F. Mi and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *Proc. of the 2015 ICDM Workshop on Data Mining for Educational Assessment and Feedback*, pages 256–263, November 2015.

[13] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé, III, and L. Getoor. Learning latent engagement patterns of students in online courses. In *Proc. of the 28th AAAI Conference on Artificial Intelligence*, pages 1272–1278, 2014.

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

[15] D. Yang, T. Sinha, D. Adamson, and C. P. Rose. "Turn on, Tune in, Drop out": Anticipating student dropouts in massive open online courses. In *Proc. of the 2013 NIPS Data-Driven Education Workshop*, volume 11, 2013.