Computers in Human Behavior 58 (2016) 119–129

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Full length article

Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization

Wanli Xing [a, *], Xin Chen [b], Jared Stein [c], Michael Marcinkowski [d]

[a] Department of Educational Psychology and Leadership, Texas Tech University, Lubbock, TX 79409, USA
[b] School of Engineering Education, Purdue University, West Lafayette, IN 47906, USA
[c] Instructure Incorporation, 6330 S 3000 E #700, Salt Lake City, UT 84121, USA
[d] College of Information Sciences and Technology, Penn State University, State College, PA 16801, USA

ARTICLE INFO

Article history:
Received 6 July 2015
Received in revised form
17 November 2015
Accepted 2 December 2015
Available online xxx

Keywords:
MOOC
Dropout
Prediction
Algorithm
Stacking
Learning analytics

ABSTRACT

Massive open online courses (MOOCs) have recently taken center stage in discussions surrounding online education, both in terms of their potential as well as their high dropout rates. The high attrition rates associated with MOOCs have often been described in terms of a scale-efficacy tradeoff. Building from the large numbers associated with MOOCs and the ability to track individual student performance, this study takes an initial step towards a mechanism for the early and accurate identification of students at risk for dropping out. Focusing on struggling students who remain active in course discussion forums and who are already more likely to finish a course, we design a temporal modeling approach, one which prioritizes the at-risk students in order of their likelihood to drop out of a course. In identifying only a small subset of at-risk students, we seek to provide systematic insight for instructors so they may better provide targeted support for those students most in need of intervention. Moreover, we proffer appending historical features to the current week of features for model building and to introduce principle component analysis in order to identify the breakpoint for turning off the features of previous weeks. This appended modeling method is shown to outperform simpler temporal models which simply sum features. To deal with the kind of data variability presented by MOOCs, this study illustrates the effectiveness of an ensemble stacking generalization approach to build more robust and accurate prediction models than the direct application of base learners.

Published by Elsevier Ltd.

1. Introduction

Online education is one of the fastest growing segments in education with one particular form of it – massive open online courses (MOOCs) – recently taking center stage in discussions of its future. A MOOC is usually "*massive*, with theoretically no limit to enrollment; *open*, allowing anyone to participate, usually at no cost; *online*, with learning activities typically taking place over the web; and a *course*, structured around a set of learning goals in a defined area of study" (Educause, 2013, p.1). These courses are most often offered through platforms such as Coursera, edX, and Udacity, which support teachers as they deploy courses that may scale up to hundreds or even thousands of students. Growing out of the Open Educational Resources (OER) movement, MOOCs are gaining popularity in large measure because they provide a specific means in order to achieve more equitable access to learning. So far, however, there is little evidence that the potential imagined for MOOCs is being realized. High attrition rates of MOOCs (ranges from 91% to 93%) have often been highlighted as a scale-efficacy tradeoff (Onah, Sinclair, & Boyatt, 2014).

While MOOCs demonstrate the potential of using the Internet to make education available to a broader base, the large number of students who enroll in (and drop out of) each course raise methodological difficulties for instructors as they work to identify academically at-risk students and provide in-time interventions. To some extent, the scaling up of learning in MOOCs can be considered as a sacrifice of pedagogical support (Brinton et al., 2013). It is almost impossible to offer the same quality of support in a class of five thousand as in a class of fifty due to the difficulty in collecting and analyzing data from such a large number of students. This situation may become worse since traditional educational

* Corresponding author.
E-mail addresses: wanlixing.la@gmail.com (W. Xing), chen654@purdue.edu (X. Chen).

http://dx.doi.org/10.1016/j.chb.2015.12.007
0747-5632/Published by Elsevier Ltd.

researchers and practitioners have been using methods such as surveys, interviews, focus groups, and observations for data collection, methods which are time consuming and limited when conducted at scale (Xing & Goggins, 2015; Xing, Kim, & Goggins, 2015). Further, such methods are unable to support timely interventions for at-risk students, which, given their high dropout rates, is a central concern for MOOCs. The emerging fields of learning analytics and educational data mining (Siemens & Baker, 2014; Xing, Wadholm, & Goggins, 2014, Xing, Wadholm, Petakovic, & Goggins, 2015) seem to offer promise in solving the dropout problems in MOOCs. In particular, learning analytics techniques and educational data mining enable analyzing the low-level trace data regarding students' interactions with a course and with other students (Xing & Goggins, 2015; Chen, Chen, & Xing, 2015). From this kind of low-level structured data, it is possible to automatically infer higher level student behavior (e.g. dropout) in order to inform educational decision-making (e.g. intervention). The automatic nature of methods based on learning analytics and educational data mining have the potential to meet the challenge of large scale in MOOCs while at the same time also satisfying the requirement for being able to support timely interventions.

However, existing studies which employ click stream data to examine student dropout in MOOCs have mostly focused on summative measures of attrition, overlooking the temporal requirement for designing and implementing intervention. By performing a correlation analysis between the course completion and trace data evidence of engagement in the course, many studies have attempted to identify factors that predict the completion of the MOOC course (e.g. Alraimi, Zo, & Ciganek, 2015; Yang, Sinha, Adamson, & Rose, 2013). Similarly, preliminary prediction research applies all-time trace data to identify which students may dropout or not. Unfortunately, these studies are unable to meet the requirement that interventions be able to be implemented early enough in a course that they are effective (Halawa, Greene, & Mitchell, 2014). By the same token, the massiveness of the number of students dropping out in a course renders prediction methods depending only on the first week or only certain points of time less effective. Even though it can detect whether students are at risk of dropping out in a timely manner (Jiang et al., 2014), this model is unable to predict exactly when the student is dropping out. That is, while thousands of students may be flagged as being at-risk after the first week, such methods are unable to indicate which ones are in danger of dropping out after only the first week or which ones will still remain active after two, three, or four weeks, only to eventually drop out of the course. Such a method, while effective at predicting all students who may eventually drop out of a course, does not support teachers in identifying those students in need of immediate intervention. In this, constructing a temporal prediction model is critical as such a model would be able to place these at-risk students in a chronological order of when they are most at risk of dropping out so that teachers can provide timely intervention to the students most at risk at any given time.

Addressing the temporal features of a prediction model is significant not only because it allows for early detection of dropout students but also because of the gradual nature of attrition in MOOCs. This gradual attrition is especially the case for students who participate in course discussion forums (Yang et al., 2013). Although a major portion of participants dropping out either never engage in course activities at all or drop out after the first week, a large fraction of participants remain in the course for several weeks only to drop out later, with such a pattern suggesting a struggle to stay involved. Such struggle to stay involved is seen most directly in those students participating in the online discussion forums. These students, taking part in a massive community of strangers, lack the kinds of shared practices that help to form supportive bonds of interaction and are easily overwhelmed by the volume of discussion present in the forums (Rosé et al., 2014). As such forums are a key aspect of MOOC platforms, students involved with the online discussion boards are more prone to stay through the course. As such, students struggling with a course and yet still engaging with the discussion forums represent low hanging fruit which may be able to be targeted in order to enhance the success rate of a MOOC (Yang et al., 2013). Given this, we propose a method for predicting the gradual falling away from participation in a course which focuses in large part on forum participants.

This kind of prediction, from a learning analytics and educational data mining point of view, is usually realized by supervised machine learning algorithms (Goggins, Xing, Chen, Chen, & Wadholm, 2015). In order to forecast students' dropout, a training set of previously labeled (dropout or not dropout) data instances is used to guide the learning process, while another set of labeled instances named the “test set”, is applied to measure the quality of the obtained model (Xing, Guo, Petakovic, & Goggins, 2015). Previous studies have applied different algorithms (e.g. logistic regression, Naïve Bayes, decision tree, etc.) to perform the prediction. However, a simple application of these machine learning algorithms directly to the MOOC data may not adequately respond to the unique characteristics of the data generated in MOOC learning platforms. Due to their openness, the data generated in MOOCs can vary significantly overtime, with the number of students dropping out or completing the course differing substantially from course to course (Brinton et al., 2013). As a result of these fluctuations and variability in the data, the performance of these algorithms can be significantly altered. Because of this, a more reliable predictive modeling mechanism is needed.

Due to the large number of dropouts from the course, the gradual manner in which they fall away from a course, and data variability of MOOCs, this work proposes a temporal prediction model using ensemble machine learning methods that aims to accurately and reliably identify struggling students in MOOCs in advance so that teachers can provide timely and quality pedagogical support to harvest these low hanging fruit of MOOC forum participants and keep them engaged in the course. Specifically, to address the immenseness and graduality of attrition, we design a temporal modeling method, through which we predict who is going to drop out next week. In other words, instead of using all the data to identify all the students at risk of dropping out, the model is able to specifically determine students at-risk of dropping out for the following week using data collected from previous weeks. In only calling attention to those students at risk of dropping out in the coming week, this temporal modeling mechanism enables teachers to focus on only that small group of students in immediate danger instead of being faced with an overwhelming number of all the students who may drop out at some point in a course. With this, the teacher can deliver greater support to a smaller number of at-risk students each week. In terms of model building, instead of simply summing features together over weeks, an appended features mechanism based on principle component analysis (PCA) is used to expand the feature space. With regard to the performance of machine learning algorithms when confronting the fluctuating dataset, this study proffers the ensemble approach – stacking generalization – to increase the prediction stability and performance.

The overall research question for this project is “How and to what extent can we build a prediction model that can accurately and reliably identify struggling students in MOOC forums in advance so that teachers can provide timely and quality pedagogical support to them?” Two specific research questions are raised based on the overall goal:

1) How can we synthesize the features for temporal model construction to improve the prediction performance?
2) How can we employ stacking generalization to improve the temporal prediction performance?

The major research goals of this study are 1) to experiment and demonstrate a temporal modeling approach for students' dropout behavior; 2) to show the advantage of appended feature modeling space based on PCA over a summed features modeling space; and 3) to explore the power of the ensemble learning method (stacking generalization) in enhancing the prediction ability. The rest of the paper is organized as follows: Section 2 presents the related studies and background information. Section 3 describes the data and context of the study. Section 4 describes the research methodology. Section 5 presents the experimental results and analysis. Section 6 discusses the results, and Section 7 concludes this study.

2. Research background

2.1. Dropout in MOOCs

In spite of their momentum, student retention remains a serious problem for MOOCs since, due to their open nature, enrollment is open to the general public and consequences for failure in a course are minimal to none. This openness results in a large portion of students registering for the course without ever actually participating in it and students continuously dropping out at virtually every point during the course (Yang et al., 2013). Even though the problem of students registering and then never participating in a course is largely outside of the purview of instructors, the gradual falling away of the number of students participating in the course is nevertheless something that may be remedied through teacher intervention (Rosé et al., 2014).

While discussion are still underway as to whether the completion rate is actually a problem indicating partial failures of MOOCs or they merely reflect the diversity of MOOC learners and their intentions, the low completion rates do raise serious questions in terms of the MOOCs' effectiveness (Chafkin, 2013; Marcus, 2013). In regular classes, students engage with course materials in a structured and monitored way with teachers directly observing student behavior and providing feedback. However, for MOOCs, the scale, heterogeneity, and distributed nature of the students requires new methods for both providing student support and guiding teacher intervention. These unique characteristics of MOOCs undermine the effectiveness of traditional methods such as direct observation or the use of questionnaires, interviews, or focus groups (e.g. Alraimi et al., 2015; Margaryan, Bianco, & Littlejohn, 2015) in understanding student dropout behavior. As has been shown by researchers and practitioners in the learning analytics and educational data mining communities, the automatic analyzation of large scale behavioral trace data provides an alternative to these methods when looking to improve student engagement, retention and outcomes (Romero & Ventura, 2010).

2.2. Dropout factors and analysis in MOOCs

Prior work has utilized correlative analysis to understand factors affecting student success and failure (as well as dropout and retention) in MOOCs. A major portion of these studies use traditional questionnaires for data collection. For instance, Gutl, Rizzardini, Chang and Morales (2014) investigated motivations for enrolling in a MOOC and discussions reasons for the attrition during the course through a survey of 134 students. Alraimi, Zo and Ciganek (2015) examined 316 survey responses in order to understand the role of openness and reputation in relation to course completion rates and student intention to continue using MOOCs. While providing a valuable snapshot of student behavior, these static data collection methods are not able to capture a dynamic picture of the factors that influence students' behaviors, particularly as they are variable over time, and only provide a partial sampling of the attitudes of those students who choose to respond to the questionnaire.

Other research has explored low completion rate through correlation analysis which takes advantage of the detailed behavioral trace data provided by MOOCs. Student behavior and social positioning in discussion forums (Yang et al., 2013), sentiment in forums (Wen, Yang, & Rosé, 2014), and peer influence (Yang, Wen, & Rose, 2014) have all been investigated as possibly playing a role in student success. Based on an analysis of trace data, Muñoz-Merino et al. (2015) have illustrated how to examine students' effectiveness in interacting with educational resources and activities in MOOCs in order to gain insight into students' performance. These studies have helped to build an understanding of the reasons for student dropout or failure, and have suggested that student dropout is predictable when relying on quantified information generated from the trace data, but none have provided an ultimate prediction model.

Other studies have gone beyond an analysis of only limited aspects of student behavior and have used machine learning algorithms employing electronic trace data to predict students' engagement or success in MOOCs. Jiang, Warschauer, Williams, ODowd, and Schenke (2014) applied logistic regression to predict students' final performance in the course based on their performance in the first week of a course. Ramesh, Goldwasser, Huang, and DaumeGetoor (2013) have used PLS logic to analyze students' online behavior and to identify two different types of engagement, which are then further employed to predict students' final performance. Kizilcece, Piech & Schneider (2013) implemented cluster analysis to identify different engagement and disengagement patterns in three computer science courses. They characterized emerging clusters of students as completing, auditing, disengaging, and sampling learners, and then provided recommendations depending on the learning trajectories in each cluster. While each individually valuable, these studies still do not directly address questions such as which students will dropout and when.

2.3. Dropout prediction model in MOOCs

So far, one possible method for increasing the completion rates of MOOCs has been to build dropout prediction models using machine learning algorithms to predict when a student will stop visiting the course depending on his or her prior behaviors. Kloft, Stiehler, Zheng, and Pinkwart (2014) have illustrated such prediction modeling utilizing low-level trace data using Support Vector Machine (SVM). Similarly, Halawa et al. (2014) built a dropout prediction model using student activity features corresponding to a potential lack of ability or interest. Balakrishnan and Coetzee (2013) applied Hidden Markov Models (HMMs) to predict student dropout using features generated from discussion forums and video lectures. Perhaps most innovatively, Taylor, Veeramachaneni, and O'Reilly (2014) employed crowd-sourced feature engineering to detect dropout relying on logistic regression. By and large, however, studies which leverage such data analytic methods do not consider the massive number of students in MOOCs and the correspondingly massive number of potential dropouts. So, while such prediction models have been demonstrated to identify at-risk students accurately and early enough (for example, using only the first week of data, a model may be able to accurately detect 9000 students who will dropout), given the volume of positive identifications, the pedagogical support a teacher can provide to these students is

limited to only a perfunctory intervention (such as an email alerting a student to keep up with the course).

Beyond concerns specifically germane to online learning, many previous applications of machine learning algorithms suffer due to a lack of concern for the kind of model stability and reliability necessary for the analysis of complex phenomena such as online learning. Machine learning algorithms, especially unstable learners (e.g. decision tree, neural network), are sensitive to data perturbation (Zhou, 2012). In MOOCs, students have considerable freedom to determine what, when, where, and how they will learn. Such freedom may lead to substantial data variance, which further causes these unstable machine learners to be unreliable and less accurate. For stable learners such as SVM or Naïve Bayes, their performance can easily become poor when the classes are imbalanced or a small training sample is utilized (Zhou, 2012). For MOOCs in particular, as more than 90% of participants will eventually drop out, the class ratio between dropout and retention is highly imbalanced. To account for this, stacking generalization, as an ensemble learning approach, is proposed to relax this data variability issue.

3. Dataset

3.1. Context

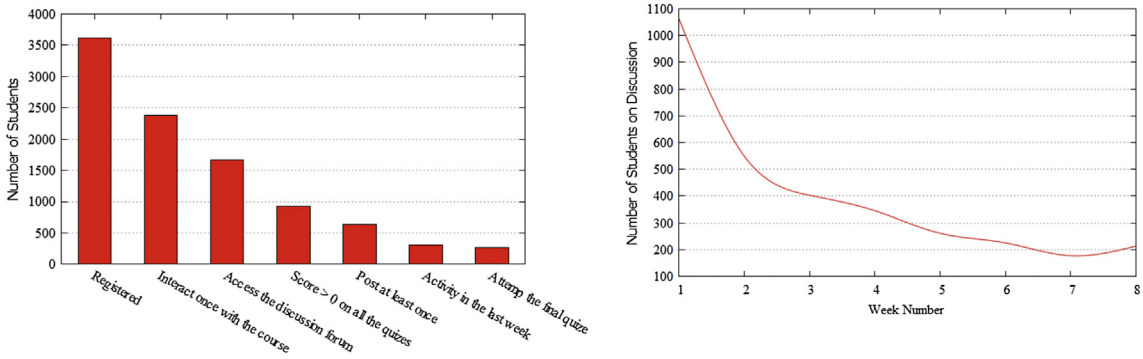
For this study we focused on a project management course launched in August, 2014, and hosted by Canvas. The course lasted eight weeks with 11 modules and 3617 registered students. Except for the first 4 modules which all took place in the first week, each module lasted roughly one week. The end of each module was in most cases accompanied by spaces for online discussion and quizzes. In total, there were 14 discussion forums and 12 multiple-choice quizzes. Due to the number of students participated in the course and discussion forum, instructors usually had limited interaction experience with students typically centered around forum interactions and brief email exchanges.

Data from this course was obtained from two sources: First, we obtained click-stream data for the entire duration of the course directly from Canvas. This contained information regarding such things as which pages students visited and when or how many times students clicked on certain sources (e.g., syllabus, modules, quizzes, etc.). Second, quiz scores and discussion forum data was obtained in Json format through the Canvas API. This data consisted of quiz scores for every enrolled student and all discussion forum content (including which students posted to the forums and when). Fig. 1 (a) provides an overview of the course as derived from the data and Fig. 1 (b) gives information on the number of students who remain active in the discussion forum during the course. In total, 1379 students participating in the discussion forum dropped out the course. It is this set of students who, with hindsight, would be considered at-risk.

3.2. Attributes

Based on data from previous weeks, the prediction model constructed is expected to be able to forecast who will dropout the following week. As such, the outcome attribute for the model is which week the student stops visiting the course – something which is easily computed using the activity stream data to determine the last date the student accessed the course. In order to construct the features for model building, the following features are computed from the click stream data for each student: Number of discussion post, number of forum views, number of quiz views, number of module views, and number of active days. So, as students may access modules several times in a single day, the number of modules views is usually larger than the number active days since the maximum number of active days in a week is 7. In addition, module view is only one action possible in the course, there are many others besides module clicks e.g. discussion forum, test. Beyond this, the social network degree is also calculated for each student as a reflection of social interaction. To construct this, each student is considered as a node and a comment from one student to another in the forum is regarded as an edge between these two students. The degree value is calculated as the number of edges the student has. Since the aim of the model is to predict whether the student will drop out in the coming week based on data from the current and previous weeks, all features are calculated for each student and for each week. Given this, later weeks have more data for model construction than the earlier weeks. Table 1 shows a complete list of features with explanations.

Fig. 2 presents a visual explanation of the data characteristics. Fig. 2 (a) shows the dropout and active students' ratio over each week. For example, for Week 3, it shows that all the students ever interacting with the discussion board until Week 3, more than 30% of student dropout, with around 65% of them still active. From a predictive modeling perspective, the class distributions between these categories over the weeks are highly imbalanced. Fig. 2 (b) shows a box plot of Week 5 for the six attributes by scaled to a unit standardization. As can be seen, they are highly skewed and with many outliers. It demonstrates the data variability in the MOOC that results from the participatory freedom given to students.



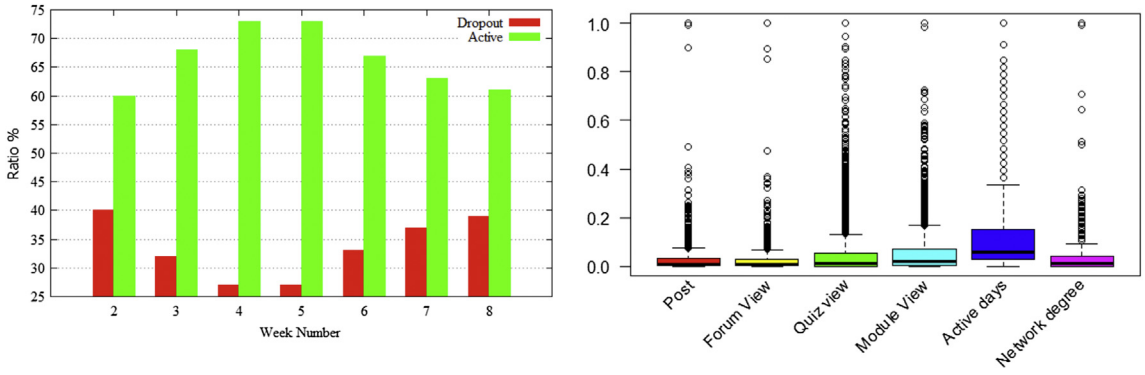
(a) Course overview

(b) Number of students interacting with the forum each week

Fig. 1. Project Management MOOC course on Canvas.

Table 1
Feature set and explanation.

Feature	Explanation
Dropout Week (predicted)	The week when the student last visits the course
Number of discussion post	Number of posts the student makes each week
Number of forum views	Number of times the student accesses the discussion forum each week
Number of quiz views	Number of times the student accesses quizzes each week
Number of module views	Number of times student accesses the module each week
Number of active days	Number of days the student interacts with the course each week
Social network degree	Number of edges the student has each week



(a) Dropout distribution over weeks (b) Week 5 features boxplot

Fig. 2. Data characteristics.

4. Methodology

4.1. Preprocess

For each week of the course ($n = 1, 2, ..., 8$), the dropout label for each student S_n, active in the current week, is calculated based on examining whether there is any activity from the student in the immediate next week and beyond. This generates the label vectors $y_n \in \{0, 1\}^{S_n}$, where 0 indicates dropout and 1 indicates active. These label vectors are then associated with the weekly feature vectors. Specifically, [number of discussion post, number of forum views, number of quiz views, number of module views, number of active days, social network degree, dropout label] are the prepared data format for each individual student to input into the machine learning algorithms, where the weekly feature vectors or independent variables (number of discussion post, number of forum views, number of quiz views, number of module views, number of active days, social network degree) are employed to predict the dropout label or dependent variable for a specific week.

Conceptually, the course is divided into 8 temporal segments, where a temporal segment lasts a week. Every student in each temporal segment has a full format of [number of discussion post, number of forum views, number of quiz views, number of module views, number of active days, social network degree, dropout label], where the model employs the feature vectors of the current week and/or with the historical (previous) weeks to predict whether the student will dropout in the next week. On the other hand, since the data are highly skewed and with lots of outliers and variability, logarithmic transformation was performed on the features set.

4.2. Temporal modeling approaches

Since the modeling can use both the features of the current week as well as features of the previous weeks to predict the potential for a student to dropout, it is necessary to make decisions regarding how to exploit this feature space to maximize the prediction performance. In this paper, we experiment with two approaches. In the first, we follow an intuitive approach, adding the values of the current features with the historical features together to generate new features for the prediction modeling. In other words, a new feature is constructed by sum the value of current week feature with previous weeks. Thus, the number of features (6) is constantly the same for all weeks as they are used to construct the model.

In the second, all the historical features in the previous weeks are directly appended as additional features to the features of the current week. That is, while the first week only has 6 features available to forecast the dropout label for the student in the second week, the second week will have 12 features to rely on to predict student dropout for the third week, with the third week having 18 features and so on. This method expands the feature space for model building and has the potential to improve the model performance. Theoretically, this method will be beneficial for the early weeks. However, as the course moves to the later stage, the feature space for model construction may become too highly dimensional, which has the risk of influencing the computational efficiency and diminishing the prediction performance.

In order to understand which of these approaches works best at various points in the duration of a course, principle component analysis (PCA) for every week was performed using both feature spaces. PCA is non-parametric method to extract the important information from confusing datasets and then represent it as a set of new orthogonal variables which are termed “principle components” (Jolliffe, 2002). It offers a direct mapping of high-dimensional data into a lower-dimensional space, while containing most of the information in the original data. PCA can also represent pattern of similarity of the observations and the variables by showing them as points in maps. This PCA method is expected to help us decide how the growing feature matrices in different weeks can be separated from each other or decide how to differentiate at-

risk of dropout students from those remaining students. A full description of PCA can found in (Jolliffe, 2002). Relying on PCA, a breakpoint was identified to inform the decision on which weeks to append the historical features and which weeks to build the model solely on the current week.

4.3. Algorithms

In this study, two algorithms – General Bayesian Network (GBN) and decision tree (C4.5) – are implemented. As an ensemble learning approach, a stacking method is proposed to augment the performance of the C4.5 and GBN base learners.

Bayesian network is a popular classifier based on sound statistical learning theory (Jensen, 1996). It consists of a qualitative part specifying the conditional dependencies between the variables and a quantitative part specifying the conditional probability of the variables (Cheng & Greiner, 2001). Formally, a Bayesian network $B = \langle X, A, \Theta \rangle$ is a directed acyclic graph, where each node $x_i \in X$ represents a variable (in our context, one of the 6 features), and each arc $a \in A$ between nodes represents a probabilistic dependency. This dependency between two nodes is quantified using a conditional probability distribution $\theta_i \in \Theta$. A Bayesian network can be applied as a classifier that produces the posterior probability of the class node given the values of other attributes. Naïve Bayes treats the class node as a special node, the parent of all the features as shown in Fig. 3 (a). By contrast, GBN considers the class node as an ordinary node, and it is not necessary a parent of all the feature nodes as shown in Fig. 3 (b). A comprehensive explanation of GBN can be found in Jensen (1996).

In addition to GBN, a C4.5 decision tree – a top-down tree growth algorithm – is also coded. C4.5 transforms a complex classification problem into a number of simple classification problems (Quinlan, 2014). A decision tree consists of a root node, a set of internal nodes, and leaf nodes as shown in Fig. 4. The root node corresponds to all the training data, choosing an attribute (feature) to divide the samples. Each value of this attribute is able to generate a branch and sample subset. Each path from the root node to the lead node in decision tree can represent a classification rule. The key to a decision tree algorithm is the selections of the node attribute value. A different selection of attribute values can generate a different tree structure. The information gain ratio of splitting is applied to calculate the attribute value in C4.5. For a more detailed account of C4.5, refer to (Quinlan, 2014).

Due to the data variability, there is risk for model reliability and model bias due to variances in training materials. In order to enhance the performance of the prediction model, stacked generalization (or stacking) is used. It is an approach used to construct classifier ensembles, in which individuals' decisions are combined to classify new instances (Dietterich, 1997). Stacking takes the predicted target classes of multiple difference base or level-0 classifiers and uses them to train a meta-learner or level-1 classifier. This meta-learner, usually a series (one for each target class) of

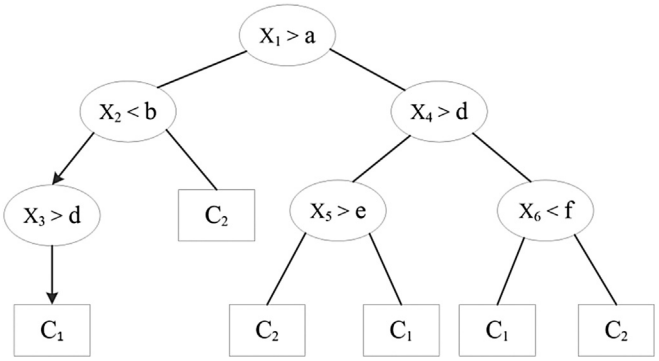


Fig. 4. Decision tree with six dimensional feature space and two classes. C_1, C_2 are the class (dropout or stay) labels. X_1, X_2, …, X_6 are the features and a, b, c, d, e, and f are the thresholds.

linear models e.g. multi-response linear regression (MLR), applies the level-0 predictions and the target classes to decide which classifiers are correct or incorrect, and finally outputs a higher level prediction relying on this. By combining the advantages of different classifiers, stacking can usually improve classification performance. A schematic diagram of how stacking works is illustrated in Fig. 5.

To some extent, this meta-learner or high level classifier can be conceptualized as the chair of a committee with the base learner as the members. Student information is first presented to the members; the chair makes the final decision on whether the student will dropout or not in the following week by considering the opinions of the members as well as the student information itself. Base learners often make different classifications or prediction mistakes. Therefore, the chair that successfully learns when to trust each member can enhance the overall performance. In this context, base learner or level-0 learner are GBN and C4.5 to compose the meta-learner or level-1 learner. A detailed explanation of stacking generalization can be found in Wolpert (1992).

4.4. Experiment and evaluation

All the modeling conducted here used cleaned and structured data subject to some pre-processing. Two sets of experiments were conducted to examine the comparison of effectiveness of prediction modeling using sum features or appended features, and the comparison of effectiveness of prediction modeling using the stacking method and with just base learners (GBN and C4.5). All these experiments were evaluated using 10-fold cross validation and 10 different runs for each partition.

To comprehensively evaluate the performance of the prediction models, the area under the receiver operating characteristic curve (AUC) is calculated. Traditional precision, recall, and accuracy are valid only for one specific operating point, an operating point normally selected to minimize the probability error (Bradley, 1997).

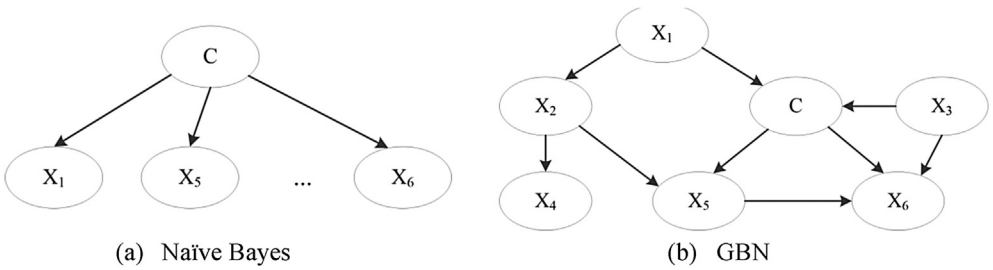


Fig. 3. Bayesian network classifiers. C is the category or dropout label. X_1, X_2, …, X_6 are the features.

Student	Features	Class
Student*1*	$x_{11}, x_{12}, x_{13}, \ldots, x_{16}$	C_1
Student*2*	$x_{21}, x_{22}, x_{23}, \ldots, x_{26}$	C_2
Student*3*	$X_{31}, x_{32}, x_{33}, \ldots, x_{36}$	C_1
Student*4*	$x_{41}, x_{42}, x_{43}, \ldots, x_{46}$	C_1
Student*n*	$x_{n1}, x_{n2}, x_{n3}, \ldots, x_{n6}$	C_2

(a)

C_1	C_2
0.84	0.56
0.38	0.85
0.62	0.66
0.48	0.77
0.89	0.28

(b)

Classifier *1*		Classifier *2*			Classifier *m*		Class = C_1
C_1	C_2	C_1	C_2		C_1	C_2	
$P_{1,C11}$	$P_{1,C21}$	$P_{2,C11}$	$P_{2,C21}$	…	$P_{m,C11}$	$P_{m,C21}$	Yes
$P_{1,C12}$	$P_{1,C22}$	$P_{2,C12}$	$P_{2,C22}$	…	$P_{m,C12}$	$P_{m,C22}$	No
$P_{1,C13}$	$P_{1,C23}$	$P_{2,C13}$	$P_{2,C23}$	…	$P_{m,C13}$	$P_{m,C23}$	Yes
$P_{1,C14}$	$P_{1,C24}$	$P_{2,C14}$	$P_{2,C24}$	…	$P_{m,C14}$	$P_{m,C24}$	No
				…			
				…			
$P_{1,C1n}$	$P_{1,C2n}$	$P_{2,C1n}$	$P_{2,C2n}$	…	$P_{m,C1n}$	$P_{m,C2n}$	No

(c)

Fig. 5. Stacking on a two-class dataset (C_1, C_2) or (Dropout, Stay) with n training data and m base classifier (GBN, C4.5). $P_{m,cn}$ notates the class prediction from classifier m for class c on example n. (a) represents the original training data; (b) is the class probability distribution; (c) is the meta training set for class C_1. Modified from Seewald (2002).

For example, a point (usually a precision and recall pair) is chosen to generate a classifier with the highest precision. However, selecting only a single operating point can cause ambiguous results when comparing two systems (Hand, 2009). AUC, as a single measure, is invariant to the decision criterion selected. Bradley (1997) conducted an empirical test on 6 real datasets and AUC shows an increased sensitivity in an ANOVA test, with decreased standard error. Besides AUC, precision is also presented in this work to show the robustness of the proposed stacking algorithm.

5. Results

5.1. Data transformation and PCA analysis

After performing the logarithmic transformation for the feature sets on both the summed and weekly features, the subsequent box plots in Fig. 6 shows that it generates fairly non-skewed distributions and the outliers are greatly reduced. Fig. 6 (a) shows the results for summed features of Week 5 and Fig. 6 (b) shows the result for features of Week 7.

To identify the breakpoint for appending the historical features for the temporal modeling, PCA is implemented to investigate the separability of dropout students with remaining students of the coming week for every week of data. As looking though the analysis results, it shows that the data become more non-isotropic in later weeks than the early stages and particularly from Week 6. Therefore, appended historical features were used in the first 5 weeks while for Week 6 and Week 7 only the current week of features was applied. Fig. 7 shows the results of Week 1 and Week 6 analysis. Another dataset was also prepared for the summed features for the historical data.

5.2. Prediction performance with sum features and appended features

Table 2 shows the results of the different modeling approaches

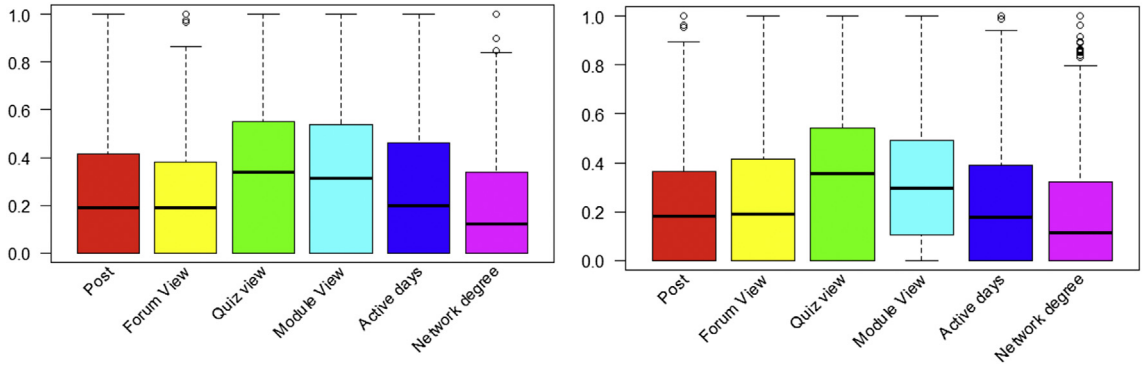
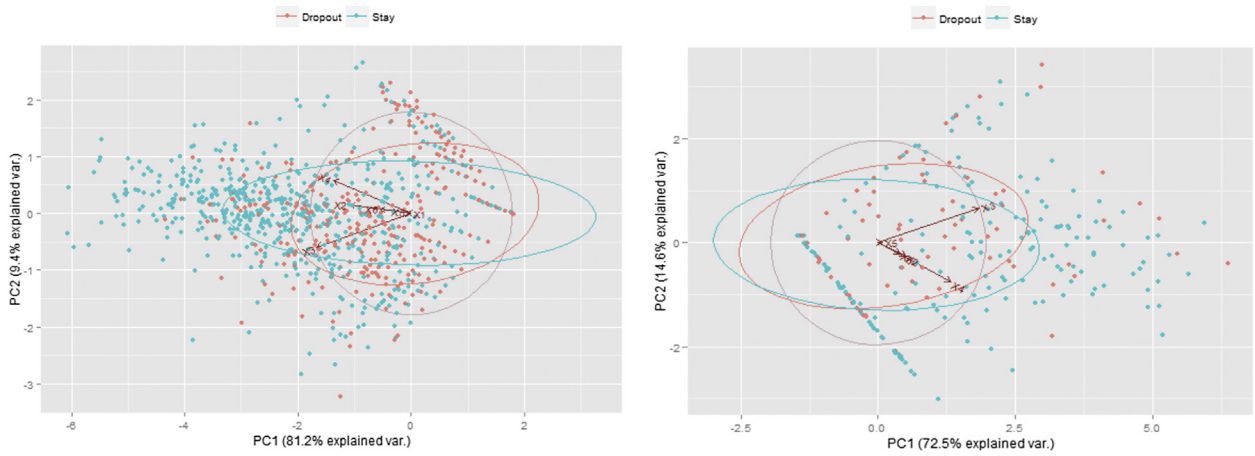


Fig. 6. Boxplots of feature sets after logarithmic transformation.



(a) PCA analysis result for the first week (b) PCA analysis result for the sixth week

Fig. 7. PCA analysis results.

Table 2
Prediction performance of GBN and C4.5 over different modeling approaches.

	GBN				C4.5			
	AUC		Precision		AUC		Precision	
	Summed	Appended	Summed	Appended	Summed	Appended	Summed	Appended
Week 1	0.802	0.802	0.793	0.793	0.805	0.805	0.772	0.772
Week 2	0.595	0.861	0.747	0.869	0.849	0.828	0.879	0.870
Week 3	0.612	0.881	0.833	0.901	0.491	0.803	0.833	0.903
Week 4	0.547	0.910	0.852	0.924	0.493	0.835	0.852	0.923
Week 5	0.654	0.908	0.837	0.908	0.494	0.893	0.837	0.921
Week 6	0.632	0.929	0.891	0.933	0.619	0.938	0.899	0.935
Week 7	0.684	0.944	0.886	0.949	0.488	0.944	0.877	0.947

over the duration of the course. The prediction performances for the first week are all the same because there are no historical features to append and they all use the same six features. The range of the AUC for GBN is [54.7%, 80.2%] and [80.2%, 94.4%] for modeling using sum features and appended features respectively. The average over the weeks for GBN is 64.7% and 89.0% for modeling with sum feature and appended features. The range of precision for GBN is [74.7%, 89.1%] and [79.3%, 94.9%] for modeling using sum features and appended features respectively. The average over the weeks for GBN is 83.4% and 89.7% for modeling with summed features and appended features. The same rule is also applicable to C4.5. The range of AUC for C4.5 is [49.1%, 84.9%] and [80.3%, 94.4%] for modeling using sum features and appended features respectively. The average over the weeks for C4.5 is 60.1% and 86.4% for modeling with sum feature and appended features. The range of precision in C4.5 is [77.2%, 89.9%] and [77.2%, 94.7%] for modeling using sum features and appended features respectively. The average of specificity over the weeks for C4.5 is 85.0% and 89.6% for modeling with summed features and appended features. Therefore, the proposed temporal modeling approach with appended features over a particular number of weeks outperforms the base line summer measures in predicting students' possibility for dropping out. This PCA analysis and the comparison of different feature spaces serves to answer the first research question regarding how we may be able to synthesize features for temporal model construction in order to improve the prediction performance.

Fig. 8, besides showing the general higher performance of modeling using appended features over summed features, also demonstrates the general trend of modeling performance. As can be seen, the performance of modeling based on summed features fluctuates over weeks and has several curves. By contrast, modeling which relies on appended features is more stable across time. The possible reason might be due to more historical data concerning the features being available for model building. In other words, the appended feature modeling method may provide additional modeling space than that given by simply summing feature values together.

5.3. Prediction performance with stacking method

This section addresses the second research question, comparing the performance of a stacking method with that of just base learners. Specifically, Table 3 and Fig. 9 demonstrate the result of predictive modeling using stacking method versing utilizing just the base learners, each using the appended feature modeling approach. The range of AUC for GBN and C4.5 are [80.2%, 94.4%] and [80.3%, 94.4%] respectively. The average for GBN and C4.5 are 89.0% and 86.3%. By contrast, when stacking these two algorithms, the range of AUC is [80.7%, 96.1%] and the average is 90.7%. Similarly, the range of precision for GBN and C4.5 are [79.3%, 94.9%] and [77.2%, 94.7%] respectively. The average for GBN and C4.5 are 89.7% and 89.6%. The precision range for stacking method is [80.7%, 95.8%] and the average precision is 91.7%. Therefore, the stacking method, which takes advantage of the two algorithms, outperforms the base algorithm alone.

Due to the high imbalance between students who remain in the course or drop out along with the data variability, it was expected that the stacking method would outperform the base learners alone, as the stacking method is known to perform well with data imbalance problems by constructing more robust prediction

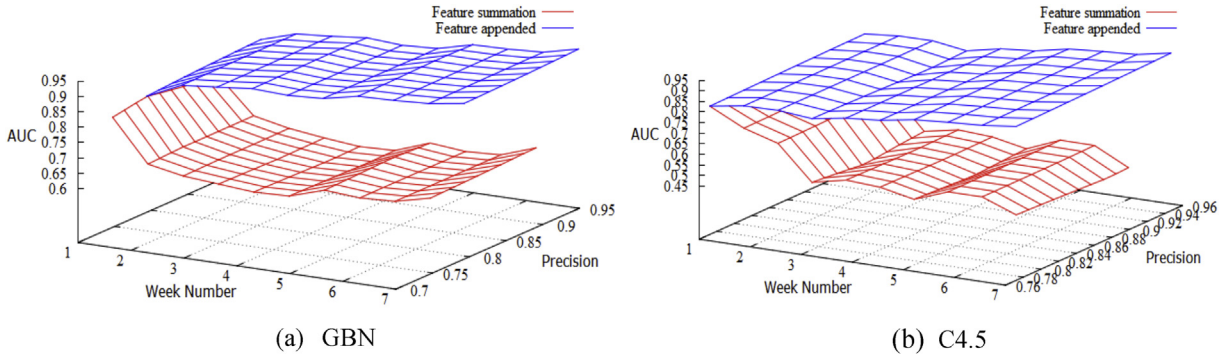


Fig. 8. Prediction performance over different modeling approaches.

Table 3
Prediction performance of stacking algorithm vs. base learner.

	AUC			Precision		
	GBN Base	C4.5 Base	Stacking	GBN Base	C4.5 Base	Stacking
Week 1	0.802	0.805	0.807	0.793	0.772	0.807
Week 2	0.861	0.828	0.877	0.869	0.870	0.897
Week 3	0.881	0.803	0.889	0.901	0.903	0.931
Week 4	0.910	0.835	0.928	0.924	0.923	0.942
Week 5	0.908	0.893	0.942	0.908	0.921	0.933
Week 6	0.929	0.938	0.942	0.933	0.935	0.949
Week 7	0.944	0.944	0.961	0.949	0.947	0.958

here, however, the instructor is able to provide support to students in a triaged fashion as shown in Table 4, with a much small number of at-risk students to worry about each week. As expected, a large portion of students drop out in the second week, therefore, teachers may need to deal with 502 students in the first week. The AUC and precision is both 0.807. But the number of students from then on decreased significantly. For example, in the third week, the teacher needs to focus on 146 students and 128 students for fourth week. The precision for both can reach 0.931 and 0.942 for the third week and fourth week respectively. These students' numbers are

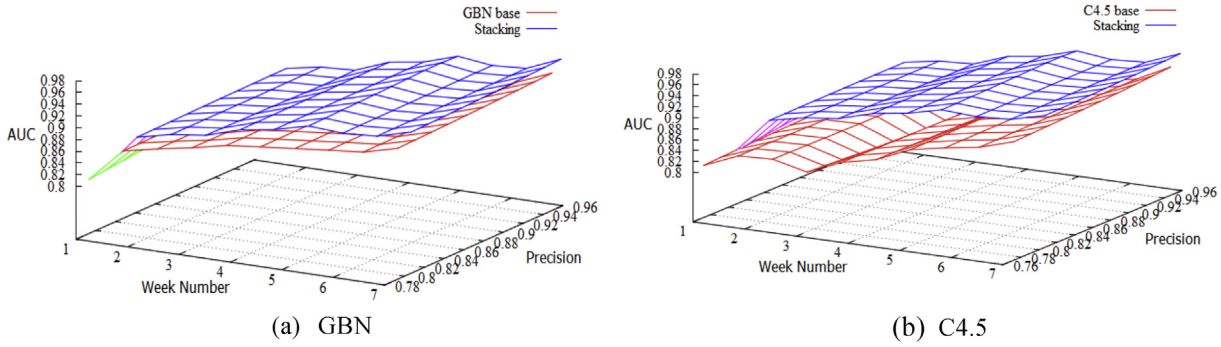


Fig. 9. Stacking prediction performance.

models. When comparing the improvement over GBN and C4.5 individually, the improvement over C4.5 is slightly more prominent than when compared to GBN. It may be because of the C4.5 is also sensitive to the choice of the node (feature) attribute value as an unstable learner, a sensitivity which a stacking approach can relax to some extent. In sum, the prediction performance can reach 96.1% for AUC and 95.8% for precision when using stacking method over appended feature space.

5.4. Intervention mechanism out of temporal prediction

For the entire course, 1667 students ever interacted with the discussion board. Among them, 1379 students eventually dropped out the course. Given the diverse and distributed nature of students in MOOCs, it is difficult for teachers to observe students' behavior and identify all of them or based on some traditional questionnaire or interview methods. Even if at-risk students are identified in a timely manner using learning analytics approaches, the instructor may still need to deal with all the identified at-risk students at one time. With the proposed temporal modeling method demonstrated roughly equal to a normal traditional college class and teachers can provide more quality pedagogical support instead of dealing with 1379 students at one time.

6. Discussion

Despite the popularity of MOOCs, completion rates for the classes are dismal in comparison with traditional online education (Brinton et al., 2013). Because of the unique characteristics of MOOCs – massive number of enrolled students, students dropping out literally at every point of the course, and the huge variability in the data produced – they raise difficult methodological questions for educational researchers and teachers as they work to provide timely and quality support for at-risk students. Traditional methods which might be used under other circumstances to identify at-risk students may not meet the unique requirements posed by MOOCs (Halawa et al., 2014).

The ability of MOOC platforms to log low level student behavioral trace data opens up opportunities for learning analytics and educational data mining methods to be used to automatically

Table 4
Student dropout over weeks and the stacking prediction results.

Week	First	Second	Third	Fourth	Fifth	Sixth	Seventh
# Student	502	226	146	128	142	129	106
AUC	0.807	0.877	0.889	0.928	0.942	0.942	0.961
Precision	0.807	0.897	0.931	0.942	0.933	0.949	0.958

identify at-risk students. However, much of the previous research using these methods has either focused on identifying factors that forecast the completion of the course or student engagement patterns (e.g. Gutl et al., 2014; Wen et al., 2014) or on simply building models to predict which student might drop out of a course (e.g. Kloft et al., 2014; Taylor et al., 2014). Simply knowing which factors influence students' attrition and who might drop out does not provide actionable intelligence to instructors as they look to provide support and intervene before a student leaves a course. Effective prediction models in MOOCs need to focus on the early detection of at-risk students. Even if some studies only use the first week of the trace data to construct the prediction model so that it can detect students' attrition early, the massiveness of the number of students may compromise the quality support instructors can offer to students.

Given the early detection requirement and massive number of students (and potential dropouts), this study proposed a temporal modeling approach, which a prediction model built using the data from both current and previous weeks in order to predict who would dropout in the following week. Through this mechanism, the dropout students are put in a chronological order through which instructors can deal with a subset of the total potential dropout students each week prioritizing the most urgent ones first. Further, instead of focusing on all the students, this work lays out an automated system which centers on working to retain the low hanging fruit of those students who are already engaged with a course in some way, as evidenced by their participation in the online discussion forums.

From a methodological and algorithmic point of view, this study experimented with various approaches toward temporal and predictive modeling. In terms of temporal modeling, rather than simply summing relevant features of student behavior in order to predict performance, this work has demonstrated the advantage of using appended features to enlarge the feature search space. Even further, in order to avoid the high dimension problem that arises in later weeks of the course, PCA was introduced and employed to identify a breakpoint for turning off the historical features. In our tests, the proposed temporal modeling method outperforms the simple using summation values of features. Moreover, where previous studies have usually applied one algorithm for prediction (Kloft et al., 2014; Taylor, Veeramachanei & O'Reilly, 2014), due to the performance impacts that result from the high variability and imbalance off the MOOC data, we introduce a stacking method. This ensemble approach allows us to tackle these data related issues, improving the performance of both the base learners and the performance of the system in general.

7. Conclusion

This study takes an initial step toward the early and accurate identification of students at-risk of dropping out of a MOOC. Specifically, this work has proposed to focus on those students already demonstrating some engagement with a course through their participation in the discussion forums. By designing a temporal modeling approach which prioritizes at-risk students according to when they are predicted to drop out of a course, we provide a mechanism by which instructors can deal with only a subset of students who are immediately at-risk on a week by week basis instead of all of them at once. This approach aims to improve the quality of intervention and support instructors are able to offer. Moreover, we proffer appending historical features to the current week of features for model building and introduce PCA to identify the breakpoint for turning off the features of previous weeks. This modeling method outperforms a simpler modeling method which relies only on summed features. To deal with the data variability and class imbalance, we utilize an ensemble stacking approach which supports more robust and accurate prediction models than the directly application of base learners.

There are several future research directions to explore: first, future studies can explore different feature engineering techniques and involve more features such as students' background information, prior experience in online learning, test grades etc. to further improve the prediction model performance; second, future work can implement this methodology on other MOOC courses and test its transferability; third, researchers can collaborate with an instructor to deploy this prediction model in actual and live MOOC courses and performed the A/B testing to examine its efficacy; last, while this model focuses on students already engaged with course discussion forums, future research can investigate how to build a more generalized model that is effective for all the MOOC students.

Acknowledgments

This work is supported by Instructure of Canvas. There is no potential conflict of interest for the work reported in this paper.

References

Alraimi, K. M., Zo, H., & Ciganek, A. P. (2015). Understanding the MOOCs continuance: the role of openness and reputation. *Computers & Education, 80*, 28–38.
Balakrishnan, G., & Coetzee, D. (May 17, 2013). *Predicting student retention in massive open online courses using hidden Markov models*. Electrical Engineering and Computer Sciences University of California at Berkeley. http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-109.html. Technical Report No. UCB/EECS-2013-109.
Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition, 30*(7), 1145–1159.
Brinton, C., Chiang, M., Jain, S., Lam, H., Liu, Z., & Wong, F. (2013). *Learning about social learning in MOOCs: From statistical analysis to generative model. arXiv preprint arXiv:1312.2159, 2013*.
Chafkin, M. (2013). Udacity's Sebastian Thrun, godfather of free online education, changes course. *Fast Company, 14*.
Chen, B., Chen, X., & Xing, W. (2015, March). Twitter archeology of learning analytics and knowledge conferences. In *Proceedings of the fifth international conference on learning analytics and knowledge* (pp. 340–349). ACM.
Cheng, J., & Greiner, R. (2001). Learning bayesian belief network classifiers: algorithms and system. In *Advances in artificial intelligence* (pp. 141–151). Springer Berlin Heidelberg.
Dietterich, T. G. (1997). Machine learning research: four current directions. *AI Magazine, 18*(4), 97–136.
Educause. (2013). *Seven things you should know about MOOCs II. Educause learning initiative*. Retrieved from http://net.educause.edu/ir/library/pdf/ELI7097.pdf.
Goggins, S., Xing, W., Chen, X., Chen, B., & Wadholm, B. (2015). Learning analytics at" small" scale: exploring a complexity-grounded model for assessment automation. *Journal of Universal Computer Science, 21*(1), 66–92.
Gütl, C., Rizzardini, R. H., Chang, V., & Morales, M. (2014). Attrition in MOOC: lessons learned from drop-out students. In *Learning technology for education in cloud. MOOC and Big data* (pp. 37–48). Springer International Publishing.
Halawa, S., Greene, D., & Mitchell, J. (2014). Dropout prediction in MOOCs using learner activity features. In *Proceedings of the European MOOC Stakeholder*

Summit (EMOOCS 2014) (Lausanne, Switzerland).
Hand, D. J. (2009). Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine learning, 77*(1), 103–123.
Jensen, F. V. (1996). *An introduction to Bayesian networks* (Vol. 210). London: UCL press.
Jiang, S., Warschauer, M., Williams, A. E., ODowd, D., & Schenke, K. (2014). Predicting MOOC performance with week 1 behavior. In *Proceedings of the 7th international conference on educational data mining*.
Jolliffe, I. (2002). *Principal component analysis*. John Wiley & Sons, Ltd.
Kizilcec, R. F., Piech, C., & Schneider, E. (2013, April). Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge* (pp. 170–179). ACM.
Kloft, M., Stiehler, F., Zheng, Z., & Pinkwart, N. (2014). Predicting MOOC dropout over weeks using machine learning methods. *EMNLP, 2014*, 60.
Marcus, J. (2013). *MOOCS keep Getting bigger. But do they work?" the hechinger report (September 12)*. http://hechingerreport.org/content/moocs-keep-getting-bigger-but-dothey-work_12960/.
Margaryan, A., Bianco, M., & Littlejohn, A. (2015). Instructional quality of massive open online courses (MOOCs). *Computers & Education, 80*, 77–83.
Muñoz-Merino, P. J., Ruipérez-Valiente, J. A., Alario-Hoyos, C., Pérez-Sanagustín, M., & Kloos, C. D. (2015). Precise Effectiveness Strategy for analyzing the effectiveness of students with educational resources and activities in MOOCs. *Computers in Human Behavior, 47*, 108–118. http://dx.doi.org/10.1016/j.chb.2014.10.003.
Onah, D. F., Sinclair, J., & Boyatt, R. (2014). Dropout rates of massive open online courses: behavioural patterns. In *EDULEARN14 Proceedings* (pp. 5825–5834).
Quinlan, J. R. (2014). *C4. 5: Programs for machine learning*. Elsevier.
Ramesh, A., Goldwasser, D., Huang, B., Daume, H., III, & Getoor, L. (2013). Modeling learner engagement in moocs using probabilistic soft logic. In *NIPS Workshop on Data Driven Education*.
Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. Systems, Man, and Cybernetics, Part C: applications and Reviews. *IEEE Transactions on, 40*(6), 601–618.
Rosé, C. P., Carlson, R., Yang, D., Wen, M., Resnick, L., Goldman, P., et al. (2014, March). Social factors that contribute to attrition in moocs. In *Proceedings of the first ACM conference on Learning@ scale conference* (pp. 197–198). ACM.
Seewald, A. K. (2002, July). How to make stacking better and faster while also taking care of an unknown weakness. In *Proceedings of the nineteenth international conference on machine learning* (pp. 554–561). Morgan Kaufmann Publishers Inc.
Siemens, G., & Baker, R. S. (2012, April). Learning analytics and educational data mining: towards communication and collaboration. In *Proceedings of the 2nd international conference on learning analytics and knowledge* (pp. 252–254). ACM.
Taylor, C., Veeramachaneni, K., & O'Reilly, U. M. (2014). *Likely to stop? predicting stopout in massive open online courses. arXiv preprint*. arXiv:1408.3382.
Wen, M., Yang, D., & Rosé, C. P. (2014). Sentiment analysis in MOOC discussion forums: what does it tell us?. In *Proceedings of Educational Data Mining*.
Wolpert, D. H. (1992). Stacked generalization. *Neural networks, 5*(2), 241–259.
Xing, W., & Goggins, S. (2015, March). Learning analytics in outer space: a hidden Naïve Bayes model for automatic student off-task behavior detection. In *Proceedings of the fifth international conference on learning analytics and knowledge* (pp. 176–183). ACM.
Xing, W., & Goggins, S. (2015). Building models explaining student participation behavior in asynchronous online discussion. *Computers & Education, 94*, 241–251. http://dx.doi.org/10.1016/j.compedu.2015.11.002.
Xing, W., Guo, R., Petakovic, E., & Goggins, S. (2015). Participation-based student final performance prediction model through interpretable genetic programming: Integrating learning analytics, educational data mining and theory. *Computers in Human Behavior, 47*, 168–181.
Xing, W., Kim, S., & Goggins, S. (2015). Modeling performance in asynchronous CSCL: an exploration of social ability, collective efficacy and social interaction. In O. Lindwall, P. Hakkinen, T. Koschman, P. Tchounikine, & S. Ludvigsen (Eds.), *Exploring the material conditions of learning: Proceedings of the computer supported collaborative learning (CSCL 2015)* (pp. 276–283). Gothenburg, Sweden: International Society of the Learning Sciences.
Xing, W., Wadholm, B., & Goggins, S. (2014, March). Learning analytics in CSCL with a focus on assessment: an exploratory study of activity theory-informed cluster analysis. In *Proceedings of the fourth international conference on learning analytics and knowledge* (pp. 59–67). ACM.
Xing, W., Wadholm, R., Petakovic, E., & Goggins, S. (2015). Group learning assessment: developing a theory-informed analytics. *Journal of Educational Technology & Society, 18*(2), 110–128.
Yang, D., Sinha, T., Adamson, D., & Rose, C. P. (2013, December). Turn on, tune in, drop out: anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS data-driven education workshop* (Vol. 10, p. 13).
Yang, D., Wen, M., & Rose, C. (2014). Peer influence on attrition in massive open online courses. In *Proceedings of Educational Data Mining*.
Zhou, Z. H. (2012). *Ensemble methods: Foundations and algorithms*. CRC Press.

Wanli Xing is an Assistant Professor in the Department of Educational Psychology and Leadership, Texas Tech University, USA with background in statistics, computer science and mathematical modeling. His research interests are educational data mining, learning analytics, and CSCL.

Xin Chen is a PhD candidate in the School of Engineering Education, Purdue University. Her research blends Social Media Data Mining and Visualization, Web Development, and User Experience Research & Design. She received a BS in Electrical Engineering from East China Normal University.

Jared Stein is the Vice President for research and education in Instructure for Canvas. He has worked in the field of technology enhanced education for more than 15 years and served as the Director of the Innovation Center in Utah Valley University.

Michael Marcinkowski is a PhD candidate in the College of Information Sciences and Technology at Penn State University. He is involved with socio-technical research pertaining to design and human–computer interaction. His main interest is in hermeneutics and the uses of empirical data in design. He is currently studying the design of online education as it exists within larger social and cultural systems.