Inferring Student Success Predictors for CS1301x Online Course at Georgia Tech

Karan Shah[1], Maxwell Bach[2], Nina Qin[3] , Amy Liu[1], Hassen Hussen[1], Ja Yoon Lee[1], Chaitanya Bapat[1], Dr. Robert Kadel[1]

1) College of Computing, 2) School of Mathematics 3) School of Industrial & Systems Engineering

Georgia Institute of Technology

Abstract

Massive Open Online Courses (MOOCs) are expanding their presence not just for online learners, but also for traditional students in universities. Georgia Institute of Technology runs an EdX (an online MOOC provider) course on CS1301x (Introduction to Python), which is an online course open to non-institutional online learners and traditional students on-campus. Using tools such as MongoDB, we aggregate and analyze the data provided by edX to determine metrics to define and predict student success. We take a set theory approach and organize students into four different classes based on intersecting features. We present results achieved on a simulated dataset based on collected data(1). Once we formalize these results, we plan on making the system modular so that it can be applied to other online courses and scaled up to larger datasets.

Overview

Research Goal

Our goal is to focus on trajectory. We want to identify any salient metrics for identifying a successful student. The variables we are taking into consideration are exercise grades and the quantity of videos watched. We plan to use these variables to understand where students lose interest, drop the course, and where additional instruction can be placed.

CS 1301x Overview

CS 1301x is a edX MOOC designed and taught by Georgia Tech professor, Dr. Joyner. In Spring 2017, the course was offered to Georgia Tech students as an alternative to CS 1301, the introduction to Python course taught in the traditional classroom setting. CS 1301x utilizes Vocareum and a proctoring software to assign students' grades. The course is taught through instructional videos and a supplementary McGraw-Hill textbook authored by Dr. Joyner. The structure of the course is broken down into 5 units. Each unit is made up of lessons. Each lesson is taught with segmented instructional videos along with corresponding multiple choice or coding exercises.

The data set we are working with currently pertains to the Georgia Tech student population who have taken the course in Spring 2017, Summer 2017 and Fall 2017.

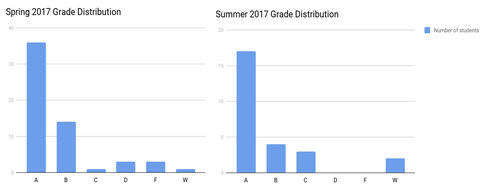

Fig 1. Grade Distribution Graphs

Both grade distribution graphs convey the success-oriented rather than bell curve distribution-oriented goal. Both are heavily right skewed. 60% of students in the Spring group received A's while 63% did in Summer. The Summer group had no failures; however the withdrawal rate did increase 2 times.

Prediction Model

For each unit, students are classified into the following classes,
Overachievers (O)
Underachievers (U)
Regular (R)
Did Not Attempt (D)
All Students (S)

Exercise Grades = $(e_1, e_2, \dots, e_n)$
Videos = $(v_1, v_2, \dots, v_n)$
Video Lengths = $(l_1, l_2, \dots, l_n)$
Exercise Averages = $(\Sigma e_1/|S|, \Sigma e_2/|S|, \dots, \Sigma e_n/|S|) = (\mu_1, \mu_2, \dots, \mu_n$ w/ standard deviation $\sigma_1, \sigma_2, \dots, \sigma_n)$
Average Amount of Video Watched =
$(\Sigma$(total amount of v_k watched)$/(|S| \bullet l_k) = (\alpha_1, \alpha_2, \dots, \alpha_n$ w/ standard deviation $\gamma_1, \gamma_2, \dots, \gamma_n)$

If student…
Within (α_k, l_k) and within $(\mu_k, 100\%)$ → student placed in (R)
Within $(0, \alpha_k)$ and within $(\mu_k - \sigma_k, \mu_k + \sigma_k)$ → student placed in (R)
Within (α_k, l_k) and within $(0\%, \mu_k)$ → student placed in (U)
Within $(0, \alpha_k)$ and within $(\mu_k + \sigma_k, 100\%)$ → student placed in (O)
Within $(0, \alpha_k)$ and within $(0\%, \mu_k - \sigma_k)$ → student placed in (D)

Toy Dataset

To test our system, we simulated a course(2) consisting of 5 units, taken by 50 students. Each student was assigned an exercise score and a video score per unit.

The exercise scores were drawn from skewed normal distributions, with the skewness determining the difficulty.

The video scores were drawn from normal distributions.

The parameters of these distributions varied by units.

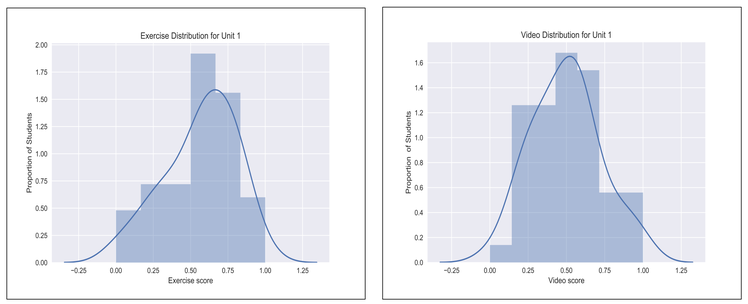

Fig 2. Exemplary distributions for a Unit in the simulated dataset

In Fig 2, we can see that the exercise distribution skews to the right. This is equivalent to a real world curve in the grades.

The video score is a normal distribution, modeled after student viewing times in the EdX data set

Preliminary Results

In the simulated dataset, the difficulty for unit 3 was set to high, which means that the exercises for unit 3 were not set properly.

Each student had an exercise score and a video viewing score for each unit, determined by the difficulty of that unit. Students were then classified based on their performance using our scoring system.

Based on this classification, we can see the number of Regular students drops and the number of underachieving student increases. On observing such a pattern, an instructor can correct the exercise to get the distribution they desire.

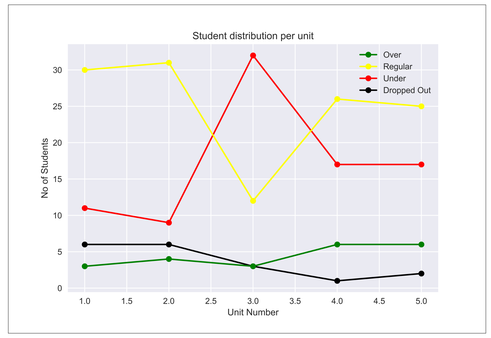

Fig 3. Category counts per unit

Future Directions

Adjusting exercises based on student performance, we can maximize set R to create a better grade distribution. Allowing the course coordinator an opportunity to see which of the offered exercises cause trouble for students in turn allows them to make micro-adjustments to the course as they see fit. It also shows outlying exercises where the content is too easy and therefore student engagement drops off. These micro-adjustments over time means future course offerings will be streamlined and these results will be replicable, eventually minimizing set O where students are not learning and minimizing set U where students are not understanding.

Using our proposed classification system, we plan to make the system modular so that it can scale up to larger data sets and be applied to the MOOC population of CS1301x. We foresee major challenges to extract large and unorganized sets from edX files, so the plan is to automate the preprocessing and storage of incoming data. Therefore, we'd be able to translate the model into code and test with real data.

References

1) Kulick, George and Wright, Ronald (2008) "Impact of Grading on the Curve: A Simulation Analysis," International Journal for the Scholarship of Teaching and Learning: Vol. 2: No. 2, Article 5.

2) Code and dataset available at: ww.github.com/karanprime/STEMExpoCode