Predicting a Student's Performance

Vani Khosla

Abstract

The ability to predict a student's performance on a given concept is an important tool for the Education industry; it allows for understanding what types of students there are and what are key concepts that help shape the understanding of another. These are important factors for educators to know in order to constantly modify and improve educating tools (such as text books and lecture plans). This paper aims to build a predictor, by finding similar student's for a given student and predicting their performance on a concept they have not explored based on past performance.

I. Introduction

Education is one of the most important industries in today's world, and while it continues to develop in meaningful ways, finding the best practices for teaching and the best evaluation tools is still a difficult problem. This project aims to predict a student's performance or score on a concept based on how other students have performed, their own performance on other concepts, and their own performance on any previous questions answered. This can naturally lead into the prediction of if a student will get a question right by using the analysis provided. This is an extremely helpful tool in the education industry to predict how a student will perform as it can help shape educational tools to provide a better learning experience. Many experiments have already been done to show that things like location, wealth, and students who are better at math and science will perform better than their counterparts, but this project aims to move away from these evaluations to find more a different result. The student specific data will not be used in this paper, and more data focused around the concepts and the specific questions will be used for features.

II. Related Work

Much of prior work done in this area of interest relies on predicting student performance based on classification and regression models. One of the most popular ways of doing so is by using student demographics (such as wealth, location, age) to create a classifier that identifies student performance. Results from several projects indicate that there is a strong correlation between student demographics and performance, but this information is not new and does not provide any new insight. A different experiment built a predictor to predict the time required for a student to answer a question and whether they would answer correctly. The features used included the student's demographics, the topic, the problem, and the time spent on the question by a student. The model was found to be very accurate, correctly predicting the time it would take for the student to submit a response and the likelihood of the response being correct. While there exists some efforts of building a recommender model—Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, and Lars Schmidt-Thieme built a recommender model that used matrix factorization that performed well based on similar student demographics—it is still a novel idea for student prediction purposes, and one such model will be explored in this paper. The model proposed in this paper will not be using matrix factorization, but instead will use similarity measurements such as Pearson Correlation Similarity to determine which students are most similar based on performance on the same concepts.

II. Dataset and Features

The dataset is provided by CK-12 Foundation, a non-profit organization whose stated mission is to reduce the cost of and increase access to K-12 education. CK-12 has provided their own material on many concepts and makes this material accessible to students for free online. At this time, several schools have been using the CK-12 material in their classrooms. CK-12 has data on student performance on practice quizzes and quizzes for many different concepts. The dataset chosen for this project has been specified below in Table 1. In order to get this into a workable format, each data point was added to a CSV file, where one row represented one data point and features as described in Table 1.

TABLE 1: The raw features provided for each data point

Test Score Id	Unique ID for a practice/quiz
Encoded Ids	Unique ID for a CK-12 concept (EID)
Student Id	Unique ID for a user
Question Id	Unique ID for a question
Level	Question difficulty level ('very easy', 'easy', 'medium', 'hard', 'very hard')
Correct	Set to 'true' if the user answered the question correctly. Otherwise 'false'.
Time Spent	Time spent by the user to answer the question
Duration	Time spent by the user on the entire practice
Created	Timestamp denoting when the practice was started

This current dataset is driven by the question id from each practice quiz. In addition to the data already provided, CK-12 has student data like grade, school, and location for many of their students. Note this data (student demographic) is not being used for the project. For this project, the data is parsed to include the student id, the encoded id (EID), and the student's score on that EID (as calculated by the number of correctly answered questions belonging to that EID divided by the total number of attempts on questions belonging to that EID).

In order to filter out any noisy data, the following constraints were applied on the data. First only EIDS part of the Biology (SCI.BIO) family were kept. Note that the EID key is set as following three letters to represent the overall topic, three letters to indicate the overall family, and digits corresponding to the concept; for example one EID found might be SCI.BIO.229. In addition all students that have only taken one practice quiz, all practice attempts with less than three questions answered, and all EIDs that were attempted by less than 100 unique students were removed from the data. Finally, for any student who has taken a practice quiz more than once the score recorded was the average of all attempts. Note that this could've been done in several different ways, i.e. only taking the student's first attempt or most recent attempt, however for this paper the average of all attempts was taken. The original dataset has 40,380,557 data points

before any filtering is applied, but after filtering dataset has 278,157 data points with 51,167 unique students and 305 unique concepts.

III. Models and Techniques

K-means Clustering:

The first thing done on the dataset was to apply *k*-means clustering to see what information could be found about student clusters. *K*-means clustering is done by assigning features to n data points (students) and partitioning into k number of groups. The algorithm uses an iterative refinement technique that works by assigning each observation to a cluster and calculating the means for each observation by finding the distance between the observation and the cluster center. This processes is iterated until the minimum means of distance calculation is attained. Any value of from 1 to the number of observations minus one is a valid k value, however for larger k values the process takes much longer as the algorithm searches for convergence. The features selected for this paper are the concept EIDS, with a score of 0 or 1 (both are attempted) assigned if the student has not taken that concept quiz.

Predictor Based on Finding Similar Students:

The main technique used in this paper is a predictor based off of a recommender system. This predictor works by finding similar students to predict what the new student's score will be for a given concept. A student is different as similar to another by comparing the concept scores for concepts taken by both students. There were three different methods used to calculate the similarity of two users Pearson Correlation Similarity (PCS), Euclidean Distance Similarity (EDS), and Tanimoto Similarity (TS). These three methods are explained below.

Pearson Correlation: (PCS)	Euclidean Distance: (EDS)	Tanimoto Similarity: (TS)
$\varrho_{X,Y} = \text{cov}(X,Y)/\sigma_X\sigma_Y$	$d(p,q) = \sqrt{(\Sigma(q^i - p^i)^2)}$	$T_S(X,Y) = \Sigma_i(X_i \wedge Y_i) / \Sigma_i(X_i \vee Y_i)$ $T_d(X,Y) = -\log_2(T_s(X,Y))$
This implementation is similar to the cosine measure similarity. The PCS is the ratio of the covariance to the standard deviation between two students scores.	This implementation is also commonly known as the root-mean squared distance. It calculates the distance between two users, where distance is defined as the square root is the sum of the differences of the first student's score and the second student's score squared. This is a fairly common method in calculating the similarity between two users.	This implementation is also commonly known as the Jaccard coefficient. The TS between two students is calculated by the ratio of the size of the intersection to the size of the union of their scores. In other words, it is the ratio of the similar scores to all their scores, or the percentage of similar scores they have.

For each of the three similarity measurements above, a model was created for the filtered dataset and evaluated. The evaluation implementation used is the average absolute difference. The score from the evaluator indicates the average deviation between the actual score and the predictor's estimate. Thus for evaluation purposes, receiving a score of 0 implies that the predictor perfectly predicted all the scores of the students in the test set.

IV. Results

The initial analysis of the dataset can be seen in Figure 1. This was the *k*-means clustering of the data by users. From first glance it doesn't appear that there is a strong set of clusters visually that separates groups of students. However, one thing to note is that for concepts that students hadn't taken a 0 was input as the score for the feature vector. This might explain why the data seems to be clustered in one area, as many students have taken a few concept quizzes but not all of them. Thus the 0 fills may have skewed the clusters. Note that this was also explored trying a -1 as the input when a student has not taken the concept, however this tended to skew the data even more, making the cluster of data points even tighter in the middle of the graph (as seen by the large portion of observations clustered in the majority section of each subfigure in Figure 1).

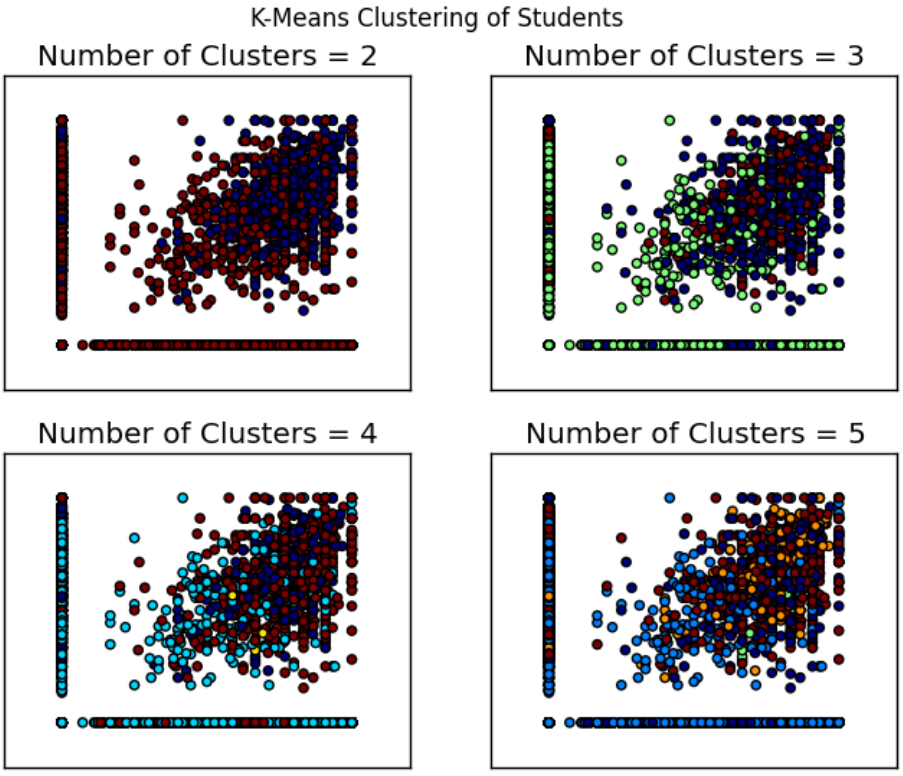

Figure 1: k-means clustering for n = 2, 3, 4, 5.

While the clustering seems to indicate that there are not strong clusters of similar students, it is important to note that this doesn't mean that is true. Clustering was only done for small groups of clusters, and it is possible that there exist smaller groups of similar students. Moving on to the

predictor phase is the only way to see how if there is strong enough data to properly predict student's performance. For reach implementation (similarity calculation) a different predictor was built and evaluated to see which had the best performance. The results from the five different implementations of the predictor are given below in Table 1. From the results, it is clear to see that the best performing predictor is the Tanimoto Coefficient Similarity predictor (or Jaccard coefficient).

TABLE 1: Results for Predictors

Implementation	Score
Pearson Correlation Similarity	0.17587
Pearson Correlation Similarity (weighted)	0.17587
Euclidean Distance Similarity	0.17510
Euclidean Distance Similarity	0.17510
Tanimoto Coefficient Similarity	0.13866

The Tanimoto Coefficient Similarity predictor had the best performance with a 13.9 % average deviation from the actual results, and the other four predictors showed fairly good results with between a 17.5 -17.6 % average deviation from the actual results. It is interesting to note that applying weighting to the similarity calculations (i.e. weighting the most similar students score higher than other similar students scores) did not show any change in the results of the predictor.

V. Conclusions

The results in this paper indicate that there are similar students and that performance can be predicted fairly well based on similar users. The students explored in this paper were selected due to their activity in the Biology family, as it is the most commonly explored family in the CK-12 dataset. It's important to note that improvement in this predictor can be made as more data is collected. Since the predictor only finds similarity using the concepts both students have performed in, the more concepts students take the more the data will be filled in, allowing for more accurate similarity calculations between students. While the *k*-means clustering didn't prove to show strong groups of clusters, that doesn't mean that those clusters don't exist. In fact the predictor's performance indicates that there are strong similarities between students and that student performances can be predicted based on past performance on different concepts. Clustering results can be explored further by changing the cluster size or finding a different way to represent a student's score if they had not taken the concept quiz. In addition more exploration of the clustering performance can be done in future work using silhouette values.

While the recommender provided in this project gives good insight, it also shapes more questions that can be explored and answered. In future work, it would be interesting to try and identify which concepts are prerequisites for others based on student performance, and which concepts are key for the understanding of an overarching category like Biology.

VI. References

[1] Beck, Joseph E., and Beverly Park Woolf. "High-level student modeling with machine learning." Intelligent tutoring systems. Springer Berlin Heidelberg, 2000.

[2] "CK-12 Foundation." Free Online Textbooks, Flashcards, Practice, Real World Examples, Simulations. Accessed December 8, 2015. http://www.ck12.org/.

[3] Kotsiantis, S., Christos Pierrakeas, and P. Pintelas. "PREDICTING STUDENTS'PERFORMANCE IN DISTANCE LEARNING USING MACHINE LEARNING TECHNIQUES." Applied Artificial Intelligence 18.5 (2004): 411-426.

[4] Kotsiantis, S., Kiriakos Patriarcheas, and M. Xenos. "A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education." Knowledge-Based Systems 23.6 (2010): 529-535.

[5] Raya, Thejaswi. "CK-12 Data." Interview by author. October 20, 2015.

[6] Thai-Nghe, Nguyen, et al. "Recommender system for predicting student performance." Procedia Computer Science 1.2 (2010): 2811-2819.

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.