Discovering the Pedagogical Resources that Assist Students in Answering Questions Correctly – A Machine Learning Approach

Giora Alexandron
Massachusetts Institute of Technology
giora@mit.edu

Qian Zhou
Tsinghua University
zhouqian@gmail.com

David Pritchard
Massachusetts Institute of Technology
dpritch@mit.edu

ABSTRACT

This paper describes preliminary results from a study in which we apply machine learning (ML) algorithms to the data from the introductory physics MOOC 8.MReV to discover which of the instructional resources are most beneficial for students. First, we mine the logs to build a dataset representing, for each question, the resources seen prior to each answer to this question; Second, we apply Support Vector Machines (SVMs) to these datasets to identify questions on which the resources were particularly helpful. Then, we use logistic regression to identify these resources and quantify their *assistance value*, defined as the increase in the odds of answering this question correctly after seeing the resource. The assistance value can be used to recommend resources to students that will help them learn more quickly. In addition, knowing the assistance value of the resources can guide efforts to improve these resources. Furthermore the order of presentation of the various topics can be optimized by first presenting those whose resources help on later topics. Thus, the contribution of this work is in two directions. The first is Personalized and Adaptive Learning, and the second is Pedagogical Design.

Keywords

Adaptive Learning, Pedagogical Design Optimization, MOOCs

1. INTRODUCTION

A central question in online courses, as in education in general, is how to design measurably more effective pedagogy. Since online courses, and specifically MOOCs, offer "full course" environments and produce log files that can be analyzed by computational tools, it is only natural that such tools would be used to optimize online pedagogy. While most pedagogic design in online education is based on 'best practices' and subjective opinions, e.g. [4], we concur with Koedinger et al. [3] that optimizing the design of instructional resources is an area in which ML and educational data mining (EDM) techniques can add a lot of value.

We propose a machine-learning, data-driven method that yields various kinds of analytics that can be used by course designers to improve their courses. Specifically, our work concentrates on computing the assistance value of instructional resources. Seaton et. al. [6] showed that the resources used for homework and exam problems differed dramatically, but did not evaluate the *effectiveness* of the selected resources. Our method aims at discovering exactly this – the contribution of particular instructional resources (e.g., page 121 in the e-text) for solving specific questions. From this, various other measures can be derived, such as which resources are generally useful, which questions do not have good supporting resources, etc.

Our longer term vision is that this can be used to augment educational resources with meta-data describing their contribution to various tasks, in line with Mccala's 'Ecological Approach' [5]. This approach suggests using ML and EDM to automatically infer the educational value of on-line resources in order to combine them to achieve educational goals. Inspired by this, Champaign and Cohen [1] presented an algorithm for sequencing educational resources based on their educational value for a specific knowledge unit. Our work suggests means for computing these values, which their algorithm takes as an input.

In the context of personalization, a lot of work has been done in predicting performance and sequencing questions, for example the interesting algorithm of Segal et al. [7]. Our preliminary results show that considering the particular educational resources that students used can also improve the prediction of their performance. This is especially relevant in MOOCs, since the students are free to choose their path through the course, and can attempt a question without going over the pedagogical resources that are important for solving it.

Our approach is based on a two-step method for computing the assistance value of instructional resources. The first step aims at identifying questions that have strong connection to the course resources. The strength of the connection between a question and its resources is operationalized as the difference between the accuracy of a prediction model that considers resources seen prior to attempting the question *and* previous performance, and the accuracy of a model that considers *only* previous performance. On such questions we conduct a second step, aiming at identifying *which* are the contributing resources and quantifying their value. The results have two immediate payoffs. One is optimizing the course design. The other is content recommendation.

The rest of this paper is organized as follows. Section 2 describes our method in detail. Section 3 presents preliminary results obtained from running the method on the Introductory Physics MOOC 8.MRev. Section 4 discusses limitations, and Section 5 presents directions for future work.

2. OUR APPROACH

This section is organized as follows. First, we define the notion of assistance value and what we consider as resources. Second, we give a high-level description of the process for calculating the assistance values. Third, we describe in more details the steps – knowledge representation, data mining, and the ML algorithms.

2.1 Resources and Assistance values

The assistance value Rq is a measure of how much a particular pedagogical resource R (say, a video explaining gravity) contributes to solving question q. It is defined as the increase in the odds that a student seeing R will solve q correctly.

The resources considered in this study are either html pages containing textual explanations, instructional videos, or questions.

2.2 High-Level Description

The process for discovering the assistance values consists of the following steps:

I. Prepare a list of the pedagogical resources from the course structure files.

II. Mine the raw data (students' logs) to create a dataset representing the resources that the students interacted with before attempting the questions.

III. Identify questions in which the resources have a significant contribution to students' success. To achieve this, we compare, for each question, the predictive power of a SVM model that considers the resources seen before attempting this question to a baseline SVM model that considers only the aggregated performance on questions attempted before this question.

IV. For each question identified in step III, discover which resources have the highest assistance value. To achieve this, we use a logistic regression with the resources as independent variables and success/failure as the dependent variable. Then, the exponents of the coefficients are interpreted as the assistance value of each resource.

2.3 Data Mining and Knowledge Representation

As first step, we build, per question, a dataset representing the resources that the student interacted with before each attempt to each question. More specifically, we use a binary feature space, with each feature representing whether a resource was seen or not. Each attempt makes an example, with '1's for the resources seen before answering, and success/failure as the binary tag of this example. Since some of the questions allow multiple attempts, a student might contribute more than one answer to a question.

We note that we chose to start with the simplest representation, and operationalized 'interacting with a resource' as a two-state condition – seen or not. We deliberately decided to use a representation that does not preserve information such as the order in which the resources were seen, the amount of time spent on each resource, and other relevant aspects of the interaction between a student and a resource, as encoding them has an exponential effect on the size of the feature space.

Performance as an additional feature. Student's ability is an important factor when it comes to predicting performance. Thus, we add it as a feature to the model. Student's ability was operationalized as percentage of success on previous attempts.

Preparing the Data. The data mining algorithm, implemented in Python, works as follows*:* For each time-sorted student log file, the algorithm scans the log while maintaining, per student, a list of the resources seen so far and an ability parameter. Each time a resource is accessed, it is added to the list (unless it is already there). Each time a question is attempted, the algorithm adds to the dataset of this question a new vector with the resources seen, the ability parameter, and a tag indicating whether this attempt was successful or not. Then the algorithm updates the ability parameter.

Exploring Various Models. To achieve the best results, we consider various models, which differ on the 'length of their memory', namely, how many resources they keep in the list. For example, a model with memory_length=5 considers only the last 5 resources seen before each attempt. Thus, for each question we actually build several datasets, one per memory_length value. The values that are considered are 1/2/3/5/10/1000. A dataset with memory_length=0 is also prepared, for benchmarking (see next section). This dataset does not 'remember' resources, only student's aggregated performance (student's ability) before attempting the question. We denote the dataset of length *j* for question *q* with *Dqj* (and omit *q* when referring to this dataset for all the questions).

The rationale underlying testing various options is mainly that we assume that some questions might require many resources, while for others, a 'long memory' might include a lot of irrelevant data.

2.4 Using SVM as a Filtering Scheme

To find questions for which the instructional resources used are significant, we train and test (using a standard 10-fold cross-validation) for each question *q* a SVM model on each of the datasets *Dqj*, for *j* = 0/1/2/3/5/10/1000 (we denote the SVM model trained on dataset *j* of question *q* with *Mqj*, and omit *q* in case we refer to this model in general). The baseline described in the previous subsection is *Mq0*. Model accuracy is measured as the average accuracy of the 10-fold cross-validation and denoted *accuracy(M)*. We then compute the relative improvement that each of the models *Mqj*, *j* = 1/2/3/5/10/1000, give over Mq0, and pick the model that gives the highest *relative improvement*, defined as $\frac{accuracy(Mj)-accuracy(M0)}{1-accuracy(M0)}$. We consider questions on which the best model gives more than 10% relative improvement as questions with strong connection to the course resources.

2.5 Using Logistic Regression to Compute Assistance Values

As described above, the role of the Logistic Regression is to identify the resources with highest assistance value for each question. This step is conducted as follows. For each problem found by the SVM to have a strong connection with the resources, we train a logistic regression on the dataset that produces the best SVM model. For example, if for a specific question *q* the most accurate model was *Mqj*, we train a logistic regression on *Dqj* (in case several SVM models give the same performance, we follow Ockham's Razor rule and take the lowest j).

The result is that per question *q*, we have a logistic model that predicts the probability of answering *q* correctly as a function of the resources seen and the ability. As described above, the coefficient attached to each feature quantifies the contribution of this feature to the final outcome, with the *p value* representing the level of confidence.

The coefficient attached to each feature is interpreted as the *assistance value* of the resource that this feature represents, and we consider only those with high confidence (defined as *p value* < 0.05).

We note that an alternative approach was to use one method both for the prediction and for quantifying the value of the resources. This approach was tried with logistic regression and with Decision Trees, which are easily interpretable machine-learning methods. However, the prediction accuracy gained by these methods was relatively low, comparing to the accuracy achieved by SVM, which on the other hand, is a much less interpretable model. Thus, we separate the process into two phases, one aims at prediction and built on SVM, and one aims at quantifying the assistance values and built on logistic regression.

In Section 5 we discuss various ways in which the prediction models and the assistance values can be used for pedagogic design and recommendation.

3. CASE STUDY – INTRODUCTORY PHYSICS MOOC 8.MReV

Context. We applied the above method on the data obtained from the 2014 instance of the introductory physics MOOC 8.MReV given by the third author and his team through the edX platform. The course attracted about 13500 students. Gender distribution was 83% males, 17% females. Education distribution was 37.7% secondary or less, 34.5% College Degree, and 24.9% Advanced Degree. Geographic distribution includes the US (27% of participants), India (18%), UK (3.6%), Brazil (2.8%), and others (total of 152 countries). (All numbers are based on self-reports.) The course lasted for 14 weeks, with content divided between 12 mandatory units and two optional ones. From the course structure file we extracted 1362 pedagogic resources (1020 problems, 273 pages, 69 videos).

Data Mining. We considered 1308 questions for which there were more than 100 student attempts. (For problems that contain several graded sections, we consider each of them as a question. Thus this number is bigger than that in the previous paragraph.) We used the logs of all the students who attempted these questions rather than restricting to those students who exceeded a particular benchmark of participation. As described in Subsection 2.3, for each question we created 7 datasets, each representing a different 'memory length'.

SVMs and choosing the questions. On the next step, we trained a SVM model on each of the datasets, i = 0/1/2/3/5/10/1000 as described in Subsection 2.4, using *R*'s libsvm [2]. This yields seven SVM models for each question, each tagged with its accuracy level. The results show that in overall, models *M1…10* performed better than *M0*, which was always at least good as majority-class prediction. This was evaluated using a paired one-side t-test that tested the hypothesis that the accuracy of *Mi* over all questions is higher than the accuracy of *M0* on all the questions, for i=1,2,3,5,10. For *M1000*, the null hypothesis was not rejected, so we cannot say that in general this model behaved better than performance-based prediction. We believe that the main explanation for this is that considering resources used long before the question at hand was even opened introduces a lot of noise into the data, reducing the weight of the proximate resources. This is exemplified in Figure 1, which shows, for 5 typical questions, the relative improvement that models 'remembering' i=1,2,3,5,10,1000 previous resources give relative to remembering only aggregated performance of each student.

Next step was to choose, per question, the best model. Figure 2 shows, per question, the relative improvement of the best model compared with the accuracy of *M0* on this question. We took relative improvement > 10% as the cut-off for defining questions with strong relation to the pedagogic resources (the line is marked in the figure). In total, of 337 questions were above this threshold.

Logistic Regression. For each of the questions identified by the previous step, we trained a logistic regression on the data that produce the best SVM model, using standard packages in *R*. For example, if for question q the best SVM was *Mi*, we trained a logistic regression for q on *Dqi*. For each question, we sorted the coefficients with p *value* <0.05 in decreasing order. This yields the assistance values. Table 1 shows an example of the two most significant resources found for a question from homework 12, which deals with gravity and orbits. According to the model, the two most significant factors that correlate with answering this question correctly are seeing the html page Angular_Momentum_of_Orbits, which explains content related to this question, and student's performance on previous questions.

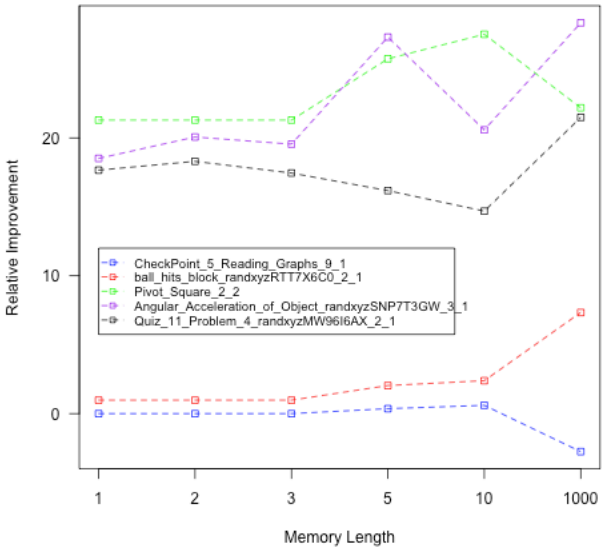

Figure 1. Relative improvement vs different memory length.

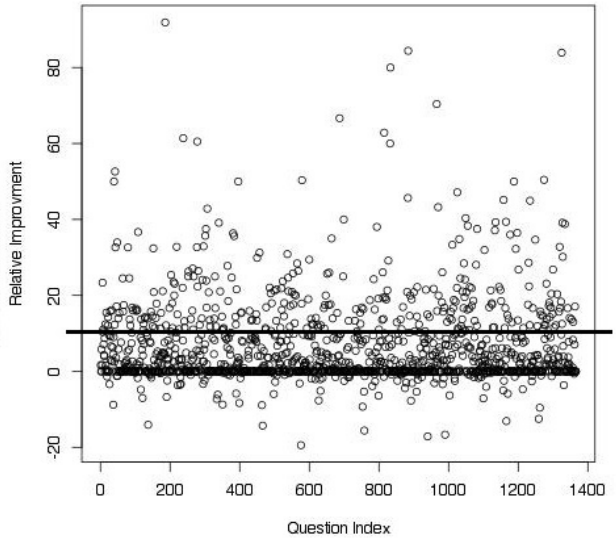

Figure 2. Relative improvement.

Table 1. Example of two most significant predictors

Question: : Homework 12, Gravity and Orbits				
Resource name	**Estimate**	**Std..Error**	**t.value**	**Pr...t..**
Angular_Momentum_of_Orbits	0.73	0.36	2.03	0.042
performance	0.62	0.2	3.05	0.003

Validation. In order to evaluate the meaningfulness of the results, we executed an expert validation protocol aimed at measuring the *precision* of the algorithm. In our case, *precision* is defined as the fraction of retrieved resources that are relevant. We gave to one of the course designers a list of 10 questions, each with 3-5 resources found to have assistance value. The course designer was asked to mark whether each resource is irrelevant/slightly-relevant/highly-relevant to the question. Weights were 0/0.5/1, respectively. In total, the precision on this sample, according to the expert, was

42.5%. We did not measure *recall*, which is the fraction of relevant resources that are retrieved, and is typically used in conjunction with *precision*, since the number of relevant resources for a specific question is unknown, and some of them can be interchangeable. Due to lack of space, we omit a detailed analysis that was done with the expert on the results given for a specific question.

4. LIMITATIONS OF THE MODEL

Our model has limitations in several areas. Due to lack of space we present them very briefly.

Cognitive. Currently the model makes simplistic assumptions on the nature of knowledge acquisition and retention. For example, it does not give any weight to the *time* spent on the resource, the *time since* seeing the resource (knowledge can be forgotten), order of resources is not considered, and it is assumed that the relation between the resources is additive (we used SVM with a linear kernel).

Model. Another limitation is that some of the independent variables in our model are collinear (i.e., A is a resource of B; A and B are resources of C). One effect on Logistic Regression is that the ability to infer the value of specific coefficients is reduced. A possible remedy is discussed in Subsection 5.2.

Data. As typically happens in real world examples, our data is skewed. For example, many of the participants already know the material (i.e., Physics teachers taking the course for professional development), so the resources they see have low effect on their ability. This adds a lot of noise to the data. Also, the ratio of examples-to-features is about 1:1, far from optimal.

5. FUTURE WORK

In this paper we described a method for computing the assistance value of pedagogic resources, presented preliminary results, and discussed limitations. Below we present directions for future work, which include further evaluation of the use of the various applications of this method, and removing limitations.

5.1 Using the Assistance Values

Finding the assistance value of resources will be useful for Pedagogical Design and for constructing Recommender engines.

5.1.1 Pedagogical Design Optimization

The assistance value can be used to address several interesting issues:

What types of resources are most effective: resources that have significant assistance value for a number of questions tell us what learning to emphasize. We can also determine the characteristics of resources that are most helpful - e.g. types (videos vs. e-text) or topics (momentum vs. energy).

Questions that lack good resources: If questions lack resources that help students to solve them this might indicate that the designer should add or improve (or possibly move closer to that question) the resources that ought to help.

Identifying redundant/bad instructional resources: If a particular resource is of little assistance for all questions, it is probably a distraction from good instruction (or covers a topic not assessed by any question).

Location of resources: Good resources that are located far from the question that they support may help students learn foundational skills.

5.1.2 Recommender Systems

In the future assistance values can be used for constructing an online resource-recommendation engine. Before a student attempts a question, the engine could use the logistic model to predict the probability that a student will get it correctly. In case this is low, a list of resources can be provided, recommended based on their assistance value, with simple metadata about each (e.g. whether e-text, a worked example, a video… as well as the median time students spent on it). This would allow the student to select the type of resource they prefer. Furthermore it would enable us to obtain much more data on the effective resources so we could determine which were best for students with different overall abilities and even possibly with different learning preferences.

5.2 Removing Limitations

Logistic regression is used both for its interpretability – to get the assistance values, and for its probabilistic classification – to predict the probability that a student will answer a question correctly. If this probability is low, we can recommend the resource with the highest assistance value that was not seen yet. One direction that we investigate is to separate this between two models – one for interpretability and another for probabilistic classification. This will allow considering other models, such as probabilistic SVMs. We note that for recommendation only, a strong probabilistic classifier is enough, and knowing the assistance values explicitly is not necessary. The process for finding the best recommendation is simple. For each unseen resource r, the engine will run the classifier on a vector consisting of the resources seen so far + r, and will recommend the resource that leads to the highest probability.

6. ACKNOWLEDGMENTS

This work is supported by a Google Faculty Award and MIT. We would like to thank Christopher Chudzicki, Zhongzhou Chen, Sait Gokalp, and Youn-Jeng Choi for useful comments, and to edX for accessing the data.

7. REFERENCES

[1] J. Champaign and R. Cohen. Ecological content sequencing: from simulated students to an effective user study, 2013.

[2] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines, 2011.

[3] K. R. Koedinger, E. Brunskill, R. S. J. de Baker, E. A. McLaughlin, and J. C. Stamper. New potentials for data-driven intelligent tutoring system development and optimization, 2013.

[4] T. L. Leacock and J. C. Nesbit. A framework for evaluating the quality of multimedia learning resources, 2007.

[5] G. McCalla. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners, 2004.

[6] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D. E. Pritchard. Who does what in a massive open online course? 2014.

[7] A. Segal, Z. Katzir, K. Gal, G. Shani, and B. Shapira. Edurank: A collaborative filtering approach to personalization in e-learning, 2014.