An Adaptive System For Standardized Test Preparation

Julia Enthoven
BS Mathematical &
Computational Sciences, 2015
jjejje@stanford.edu
CS 229

1. INTRODUCTION

How can software improve test preparation based on a student's past performance? By applying supervised machine learning algorithms, I designed and implemented an adaptive standardized test prep platform that emphasizes instruction on the concepts and skills (rule nodes) where the student is weakest. With questions organized around a set of testable topics, the adaptive engine estimates a student's proficiency with maximum likelihood techniques and draws questions from conceptual areas where the student is weakest. Applying the algorithms to data from large populations will also show which topics are the most difficult for students to pick up or remember, which further informs the question engine. I designed the system to be content agnostic so that it could be used with any standardized curriculum and expand access to adaptive learning tools in a limited-access, high-return industry.

2. DATASET

For test and train data, I implemented and added content for quantitative section of the Graduate Record Exam (GRE). The system's content database is composed of questions and explanations are taken from practice tests available on the ETS website. Similarly, the 103 procedural and fact-based rules are taken from the standards published by ETS. For each row, I extrapolated from the question and explanation which rules apply to that question.

The student learning database (SLO) tracks learner responses. For the ith learner, the student learning database (SLO) starts off empty and expands by a row after each question attempt. Each row of the design matrix designates a question/answer pair, where $X_{tj} = 1$ if the t^{th} question employs knowledge of rule j and 0 otherwise. The binary response variable represents a correct/incorrect answer.

3. MODELS

The model estimates a student's "ability" or proficiency on a set of procedural- and fact-based rules as a function of their well-correlated responses to test questions. The engine then employs this parameter to predict the likelihood that the student will answer a question that depends on the jth rule correctly:

$$P(Y = 1|\theta_j) = (1 + \exp(-\theta_j))^{-1}$$

The Effect of Forgetting:

The literature evidences the importance of recency on memory and learning [1]. In an adaptive learning context, if a student has mastered a skill then his recent responses are likely to contain correct answers. I studied two models that give emphasis to more recent results: weighted logistic regression and recent performance factor analysis (R-PFA). Linear weights multiply the effect of a question in proportion to its distance from time t:

$$P(Y_t = 1|\theta_j) = (1 + \exp(-w_t\theta_j))^{-1}$$

In contrast, R-PFA includes only results within a recency window and gives exponentially more weight to those nearest to time t. The exponential weights are multiplied by the average rate of success on past questions with rule j to get the

exponentially weighted moving average (EWMA) at time t. *a* is a tuning parameter used to promote or demote the importance of recency; following convention in the literature [2], I set a=.2:

$$EW_{jt} = \sum_{p=-2}^{t-1} \frac{a^{(t-p)} Y_{jtp}}{a^{(t-p)}}$$

This formula for the EWMA reduces noise when the student first starts answering questions by including a few "ghost" responses prior to the student's first guess, assuming that the student does not know the rule node before answering a question (Y_{11}-2=Y_{11}-1=Y_{11}). Once calculated, the EWMA is used as an additional feature on the design matrix, which is incorporated in logistic regression.

$$P\left(Y_t = 1 \middle| \theta_j\right) = (1 + \exp(-\theta_j + \delta EW_{jt}))^{-1}$$

In addition to the EWMA, I examined the effect of including the total number of questions attempted for rule j, the total successes, and both as features. These results may indicate the effect of seeing a question in comparison to answering a question correctly as predictors of a student's performance.

The Effect of Guessing:

On a multiple-choice test, a correct answer may indicate either proficiency in the question's rules or a successful guess. In reality, a correct answer may be a combination of competence and guessing; a student might eliminate answer choices based on his knowledge of the context and choose from among the remaining options. This pseudo-guessing parameter called c represents the probability of a guess leading to the correct answer and can be estimated with 1/(number of answer choices). Since my testing focused on preparation for the GRE, c=1/5.

By applying the law of total probability, I incorporate this into our logistic regression model:

$$P(Y = 1|\theta, c) = c + (1 - c)\left(1 + \exp(-\theta + \delta EW_{jt})\right)^{-1}$$

4. RESULTS

When tested on a student's response history (training set=50 questions, test=25), the basic model – along with an SVM and a Naïve Bayes calculation - performed only slightly better than chance (MSE=.44). The linear weighted model did much better, correctly predicting 88% of student responses to future questions. The two parameter-model, which incorporates the effects of guessing, performed slightly better but still with high bias.

(MSE)	SVM	Naïve Bayes	Logistic Regression (unweighted)	Logistic Regression (weighted)
One-parameter Model (train=24, n=3, test=10)	0.44	0.44	0.44	0.11
Two-parameter model, c=1/5	0.44	0.37	0.22	0.11
RPFA (with total attempts)	0.32	0.42	0.30	-
RPFA (with total successes)	0.14	0.42	0.16	-
RPFA (with total successes and total attempts)	.10	0.18	.14	

Recent performance factor analysis outperformed both of the simpler models. When both the total number of attempts and the total number of past successes (for student i in the jth category) were included as features, the predictions on whether or not a student would answer the next question correctly rose to 86% accuracy. On the same features, an SVM categorized 90% of future questions correctly.

The addition of total successes as a feature greatly improve the model's accuracy from .32 to .10 MSE. The model that included both total successes and total attempts as features had the lowest MSE.

5. DISCUSSION

These test results show that using an SVM training algorithm based on the results seen up to time t is more accurate than a Naïve Bayes estimate or basic logistic model. This SVM classifier can be used both as a predictive tool and as a proxy for estimating the proficiency of a student. With 90% accuracy, the SVM model can identify problems that a student is likely to answer incorrectly, focusing their study on weak areas.

The results also show the importance of including both the number of questions a student has answered up to time t and the number answered correctly as features. According to the R-PFA model, both questions answered and questions answered *correctly* show a statistically significant correlation with the likelihood that a student has mastered the underlying material.

Nonetheless, there is still a significant amount of bias in the system. To fulfill its potential and provide meaningful feedback, the adaptive model must give accurate estimates of a students' competence. The above results may be improved by incporating an additional parameter representing the difficulty of each problem. The analysis above treated the probability of the student answering two question with the same rule requisites with the same probability. However, as any test taker knows, questions can vary in difficulty regardless of their content. The relative difficulty of a problem may be due to the mix of its answer choices, its wording or presentation, its numeric values, or a number of other hidden variables.The current system does not have the capacity or dataset to estimate B_j, but I do hope to implement it in future iterations of this adaptive system TestDay system. B_j is updated according to the percentage of studiers in the dataset who have answered the question correctly.

Finally, a broader question/answer dataset is needed in order to test effectively. The question-rule mapping I have now has too few questions, making logistic regression less useful.

6. PLATFORM DESIGN (& FUTURE)

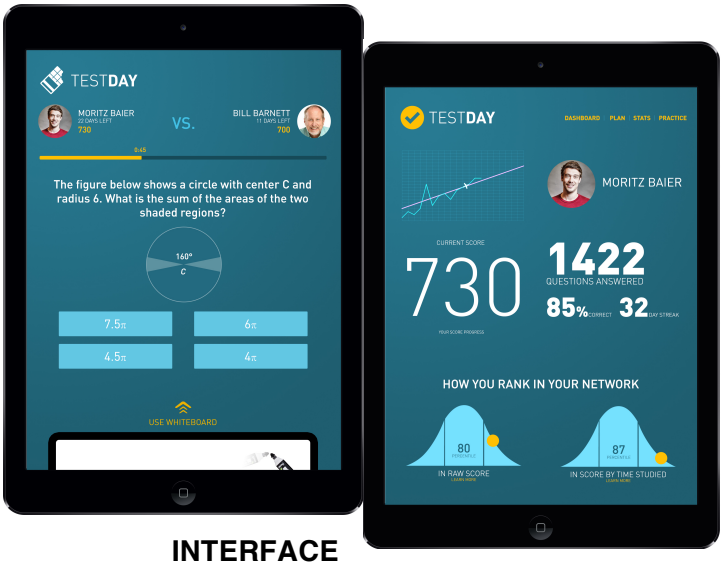
In addition to implementing the predictive models, I worked on the design of the adaptive learning system. I built a content management system, which maintains and delivers content (questions, answer choices, and explanations), stores the correct answer to determine $y^{(i)}$, and maps questions to rules; a student learning database that stores the time-stamped student input; the adaptive engine that delivers content according to a student's estimated proficiency; and a reporting dashboard.

Content Agnostic

The platform requires input of n procedural- and fact-based rules representing the content standards, a mapping of questions to those rule nodes, and the questions, answers, and choices themselves. Beyond that, the adaptive engine function without user maintenance to help prepare students for standardized exams. I hope that eventually the curriculum input can be crowd-sourced, so that new questions are submitted regularly and so that these educational materials, often guarded by test-prep companies, become accessible to everyone.

Visualization

When students have more information about their proficiency in content categories, they can be more effective studiers. Dashboards also give insight to teachers and parents, who may want feedback about how a student is performing in relation to standardized curriculum. Using javascript and photoshop, I designed the adaptive system's interface for questions, explanations, and visualization of progress on content categories. Although the data is, at this point, static, I hope to incorporate the above results to make it adaptive to student answers and interaction



INTERFACE DESIGNS

7. CONCLUSION

Adaptive learning systems have the potential to expand access to effective instruction and valuable test prep resources. If applied efficiently and accurately, machine learning can help students identify their competence on core skills and motivate them to focus on their weak areas. Moreover, algorithm-based instruction systems are smarter than conditional technologies because they are more dynamic; a student can switch between questions and topics without losing their data track. Making a content-agnostic platform would enable teachers to use the power of machine learning in any subject, making education more meaningful and impactful for students.

8. REFERENCES

[1]Baker, Ryan et al. “Detecting Learning Moment-by-Moment.” *International Journal of Artificial Intelligence in Education*, 21 (1-2). http://www.columbia.edu/ ~rsb2162/BGH-IJAIED-v29.pdf

[2]Galyardt, April and Ilya Goldin. “Recent-Performance Factors Analysis,” *7th International Conference on Educational Data Mining*. Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B.M. (eds.) http://educationaldatamining.org/EDM2014/uploads/procs2014/posters/87_EDM-2014-Poster.pdf

Falakmasir, Mohammad et al. “A Spectral Learning Approach to Knowledge Tracing.” *Conference’10*, Month 1–2, 2010. http://www.cs.cmu.edu/~ggordon/falakmasir-etal-spectral-kt.pdf

Hu, David. “How Khan Academy is using Machine Learning to Assess Student Mastery.” (Nov 2011) http://david-hu.com/2011/11/02/how-khan-academy-is-using-machine-learning-to-assess-student-mastery.html

Thorpe, Geoffrey L. and Favia, Andrej, "Data Analysis Using Item Response Theory Methodology: An Introduction to Selected Programs and Applications." (July 2012) *Psychology Faculty Scholarship*. Paper 20. http://digitalcommons.library.umaine.edu/psy_facpub/20