Machine Learning Based Directed Self Study in Calculus with CalcTutor

Kristian Kime
Computer Science Department
Brandeis University
Waltham, MA 02453, USA
Email: kristian@brandeis.edu

Timothy Hickey
Computer Science Department
Brandeis University
Waltham, MA 02453, USA
Email: tjhickey@brandeis.edu

Rebecca Torrey
Mathematics Department
Brandeis University
Waltham, MA 02453, USA
Email: rtorrey@brandeis.edu

***Abstract*—In this Innovative Practice, Work-in-Progress paper we present a new version of our web-based problem solving learning environment called CalcTutor. The key feature of the new CalcTutor system is that it organizes Mathematics problems based on the skills that they require and uses a machine learning technique (Performance Factors Analysis) to provide a difficulty estimate for each problem, personalized for each student in the system. This facility allows students to effectively engage in self-study by searching the library for problems that require specified skills. It ranks the problems found in the search by their difficulty for that particular student (the probability the student will solve them correctly on the first attempt). The current version of CalcTutor makes creating problems easy as well. It provides a simple and powerful question authoring tool which allows teachers to build both multiple choice questions and questions where the answers are mathematical functions of one variable.**

I. Introduction

This paper reports on our updating of the CalcTutor system[1] to take advantage of the work we did in creating a Student Dashboard[2] based on Performance Factors Analysis (PFA)[3]. The CalcTutor is an online educational tool along the lines of WeBWorK[4] but with several key additions. It allows easy creation of machine-verifiable questions with automatically generated difficulty estimates, personalized to each user. These questions can then be used either in a standard quiz style or as a self learning system where students search for questions of the right difficulty level that contain the skills they want to work on.

The initial version of CalcTutor had simple, one dimensional, hand crafted functions that produced the difficulty scores for the questions. In order to make our difficulty metric data-driven, more accurate, and more versatile, we looked into statistical techniques and in particular Performance Factors Analysis (PFA)[3]. This technique is based on using domain expertise to create a set of skills which are then associated with the questions (each question is labeled as requiring a set of skills). Student answer history combined with the question skill information allows us to generate coefficients for skills that can be used to predict the probability that a student will get a question correct.

In our previous work, we expanded on the basic PFA framework by proposing a method to use its skill coefficients to create a student dashboard. This dashboard is a way to measure how well a student is doing in each of the skills. While the numbers are a little hard to interpret in isolation, when we compare a student to a group, say the student's classmates, it becomes easy to see where a student is in relation to that group. The current Dashboard has been redesigned to focus on the student's placement relative to the context group, which could be the current class of students, or a reference group of students who have already completed the course, which allows them to see when they have achieved relative success. The Skill Dashboard is similar to what Intelligent Tutoring systems might call a "Student Model" – a way to keep track of what the student knows. The difference here is our design is much simpler. We cannot describe the student's level of understanding in as much depth but it is easier to create and understand.

II. Previous Work

The initial version of the CalcTutor had a reasonably robust capability for creating and answering questions, but it also came with several limitations. We didn't have a general method for creating questions; instead there were several fixed question types. For example users could create "derivative questions" where they entered a function of x and then students were asked to find its derivative. The system was capable of symbolic differentiation and functional equivalence checking so the creator didn't need to provide the derivative and the answer could be checked automatically. Another type of question was based around finding the tangent line to a function at a point and another around finding the roots of a function. All of the questions had custom interfaces for building and answering the questions.

Questions were also automatically given a difficulty. However this difficulty was based on heuristics we came up with by hand. While we validated that these heuristics were loosely correct they were very rough, one dimensional and not data driven.

The initial version of the CalcTutor had two ways a student could be asked a question. Teachers could create a quiz composed of any number of questions and present it to a class they were running. Alternatively we incorporated a peer learning system where students would be paired together and encouraged to ask each other "good" questions. The details of

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

that are beyond the scope of this paper but the work is based on the Teacher's Dilemma[5] and is covered in our previous paper[1]. This version of the system kept track of how well students were doing based on points students accumulated by answering questions. The harder the question the more points the student got. While this was a reasonable first pass at tracking success these points simply accumulated over time. There was no effect for answering questions correctly. So a teacher attempting to track the status of their students didn't have much support.

III. PFA Background

For this version of the CalcTutor we wanted to be able to provide richer information about the difficulty of questions and student capability. There are many techniques that can be used to measure student capability and problem difficulty, Item Response theory[6][7] being one of the most prominent ones. We decided to use a technique called Performance Factors Analysis [3] (PFA). PFA is based on Knowledge Tracing systems like Learning Factors Analysis [8]. PFA requires each question to be tagged with a set of skills. It then uses students' answers to questions tagged this way to quantify each of the skills into a set of coefficients. These are used to provide estimates for how much better (or worse) we should expect a student to be at questions with that skill, given that they've answered previous questions with that skill right (or wrong).

The key idea behind the PFA approach is to annotate every problem with the set of skills required to solve the problem and then to keep track of the history of each student's success and failure on previous problems. By using logistic regression on this data we can estimate coefficients for each skill, and then use that information to predict whether a student will be able to correctly answer a new question, based on the skills it requires and their previous history on problems involving those skills.

If there was only one skill, then the PFA would give us three numbers $\gamma > 0$ and $\rho, \beta \in [-\infty, \infty]$ with which we could predict a student's likelihood for answering the next question based on their history of answering previous questions using the formula

$$p = 1 - \frac{1}{1 + e^{(\beta + \gamma S + \rho F)}}$$

where S is the number of previous questions they answered correctly and F is the number they've answered incorrectly. If they answered $N = S + F$ questions in total, then the probability PFA would predict for the next question is

$$1 - \frac{1}{1 + e^{(\beta + \gamma N - F(\gamma - \rho)}}$$

which approaches 1 rapidly as $N \to \infty$ provided $\gamma > 0$ and F stays bounded. Thus, the PFA assumes that repeated success results in an exponentially rapid convergence to probability 1 for answering the next question but that increase is tempered by the number of incorrect answers F, and the parameters β, γ and ρ are found by logistic regression on the performance on some sample of students on a similarly tagged set of questions, not necessarily the current students or the current questions.

Here is a formal description of the method when we have multiple skills. The idea is the same, but we compute the value for each skill separately and then sum them. Let A_k be the set of skills needed to answer the kth question Q_k and let s_{ij} and f_{ij} be the number of times that student i has succeeded (s_{ij}) and failed (f_{ij}) in answering questions requiring skill $a_j \in A_k$. The PFA model is a logistic regression which finds the best fit parameters $\beta_j, \gamma_j, \rho_j$ such that the probability that student i will correctly answer question Q_k requiring the skills A_k is given by

$$p_{ik} = 1 - \frac{1}{1 + e^{m_{ik}}} \texttt{ where } m_{ik} = \sum_{j \in A_k} \phi_{ij}$$

and ϕ_{ij} is the success factor for student i and skill j:

$$\phi_{ij} = \beta_j + \gamma_j s_{ij} + \rho_j f_{ij}.$$

In this model γ_j is constrained to be positive and ρ_j is usually negative. Thus, the system will tend to predict a higher probability of success as the number of successes s_{ij} increases, provided it exceeds the number of failures f_{ij}. The constant β_j represents the inherent difficulty of a problem which contains this skill.

It should be noted that PFA makes the strong assumption that the set of skills is "correct" and that the questions are coded with this skills properly. For our current version of CalcTutor, we used a single domain expert who created the set of skills based on their understanding of that course and tagged the questions appropriately. Clearly the quality of the skill selection and tagging should have a major effect on the accuracy of PFA, but our validation step showed that this particular set of skills and problem tagging was sufficient to get high prediction accuracy. In our previous paper [2] we studied the answers of students in a WebWork system to questions that had been manually tagged with skills in order to validate our PFA setup and obtained very accurate predictions.

IV. Question Authoring in CalcTutor

Previously the CalcTutor had to have very specialized questions since we allowed students to create problems and had to build custom difficulty functions. With the addition of the skills we now need domain experts to create the questions but we can vastly expand the scope of the questions. For the new version of the CalcTutor we rebuilt the question creation system. Questions now have an introduction and any number of sections. Each of these sections can ask two basic types of questions that cover a wide variety of real world problems. The first type is multiple choice. The creator can input any number of options and indicate the correct one. The other type has an answer that is a function of x. The builder can specify any text they want that indicates what function the student should answer. For example they might ask "What is the derivative of $2x$?" or "If a particle has acceleration 2, starts with a velocity of 3 and starts at position 5 where will it be after x seconds?".

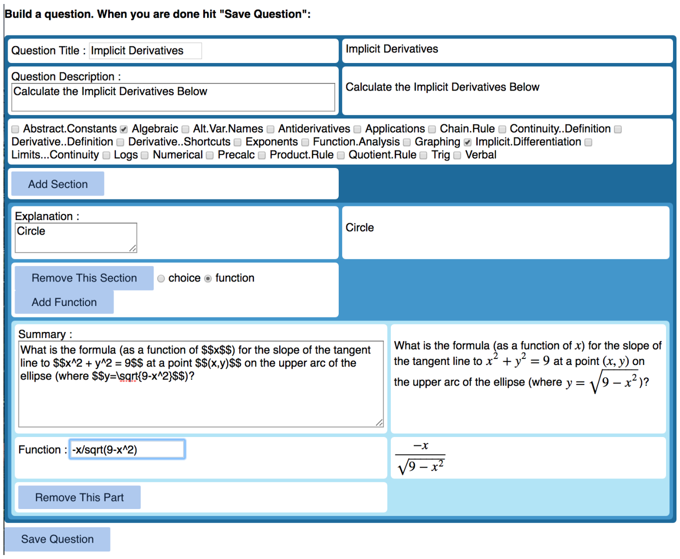


Fig. 1. Instructors view for creating CalcTutor questions

The system is able to compare the student's answer with the one provided by the creator of the problem.

The creator of the problem also needs to specify which skills the problem requires. This is as simple as checking a box for each of the predefined skills. However this is obviously a very important step as misclassifying questions will mean they do not show up when students are searching for them and the prediction of difficulty level will be inaccurate. Currently we have stuck with questions that were used in the calibration of the PFA parameters as these have obviously been the most vetted. But we plan to expand the available set. After new questions are created we plan to monitor how students answer them and recheck the PFA analysis to ensure that the system is still producing sensible results.

Fig. 1 shows a view of the question creation interface. The question author must specify the title and description of the question and then must check the boxes of the skills that this question requires. They must then create one or more sections which are either multiple choice questions or questions that require a function of one variable as an answer. The user then inserts their specification on the left side of the form and a LaTeX rendered version of their text appears on the right. The system allows the instructor access to the standard LaTeX Math syntax. The author can also specify graphs of single variable functions.

V. Question Difficulty and PFA

As mentioned before the previous version of the CalcTutor had several specific types of questions and relied on hand built difficulty metrics for each type. While this did indeed have some correlation with the chance students got the question right it was not highly accurate and was difficult to improve. PFA provides us with a much more rigorous definition of difficulty than we had before. Now we can use the question's skills and the user history to predict a probability that the student will get a given question correct. While this system does require a more advanced user to create questions it means the tuning of the actual difficulty can be automated.

VI. Student Dashboard

An additional benefit of this system is that we can combine all the information we have to give an assessment of how a student is doing in each of the skills. This requires a few steps but the basic idea is to create a fictitious question that only has one skill and use PFA to predict the likelihood that a given student is going to get that question right. We can do the same with other students in the system and see how the original student compares. This gives us a way to say (compared to a given group) how well a student is doing in any particular skill. For example we might compare a student to all the other students in a class and see that they are in the bottom ten percent at the graphing skill. Alternatively we could look at a student and compare them historically against other students in past years to see when they have achieved a reasonable level of mastery for any particular skill.

We created a Student Dashboard by running this comparison for each of the skills and plotting this on a graph showing where the student ranks compared to a given group for each skill. The idea is to be able to show a student (or a teacher) a quick summary of the areas in which they are ahead or behind the class. Figure 2 shows an example of what the dashboard looks like in the current system. All of the skills are laid out on the vertical axis. Each of these has a horizontal box and whisker plot that shows the range of skills for the group. The circles indicate outliers. The student in question then has a triangle which indicates where they fall in that group. For example if we look at the "Trig" skill we can see that this student is on the right side of the box plot indicating that they are in the top 25% of the class for this skill.

VII. Personalized Searchable Problem Library

All of these new features: skills, difficulty, dashboard, provide us with ways to support students in self study. The dashboard allows a student to see how well they are doing in each skill relative to other students in the course, or some other predetermined group. This allows students to easily identify which skills they are falling behind on and thus what areas they might want to work on.

However even when a student knows which skills they need to work on it can still be unclear what question they might want to pick. The difficulty level determined by our system can then help a student narrow down the list by directing them toward questions that are of an appropriate difficulty.

To make using these features easier we built a Searchable Problem Library. The library allows students to search for questions based on the skills required to successfully solve the problem. The search form allows the user to specify required skills and banned skills. It will then find all questions in the library that are tagged with all of the required skills and don't have any of the banned skills. It will also sort the questions from easiest to hardest using the personalized PFA prediction

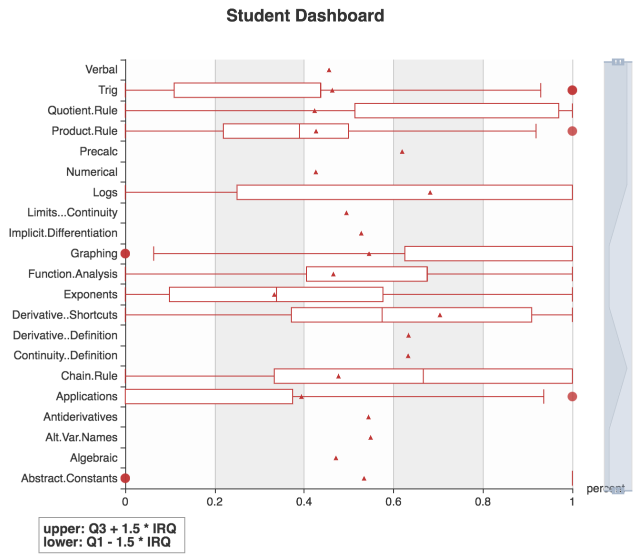
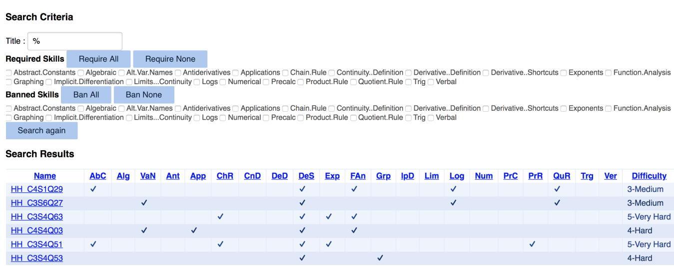


Fig. 2. Student Dashboard. This dashboard shows the student's mastery relative to a focus group which could be the current students or more aspirationally, the ranges for students who had previously completed the class. The triangle represents the student's mastery of the skill relative to the rest of the class whose mastery is indicated by the box and whiskers plot. The box is the middle two quartiles and the circles represent outliers. Skills for which no problems have yet been answered place the user at a point representing the relative difficulty of that skill as estimated by PFA.



Search Criteria

Title : %

Required Skills Require All | Require None

Abstract.Constants Algebraic Alt.Var.Names Antiderivatives Applications Chain.Rule Continuity..Definition Derivative..Definition Derivative..Shortcuts Exponents Function.Analysis Graphing Implicit.Differentiation Limits...Continuity Logs Numerical Precalc Product.Rule Quotient.Rule Trig Verbal

Banned Skills Ban All | Ban None

Abstract.Constants Algebraic Alt.Var.Names Antiderivatives Applications Chain.Rule Continuity..Definition Derivative..Definition Derivative..Shortcuts Exponents Function.Analysis Graphing Implicit.Differentiation Limits...Continuity Logs Numerical Precalc Product.Rule Quotient.Rule Trig Verbal

Search again

Search Results

Name	AbC	Alg	VaN	Ant	App	ChR	CnD	DeD	DeS	Exp	FAn	Grp	IpD	Lim	Log	Num	PrC	PrR	QuR	Trg	Ver	Difficulty
HH_C4S1Q29	✓								✓		✓				✓				✓			3-Medium
HH_C3S6Q27			✓						✓						✓				✓			3-Medium
HH_C3S4Q63						✓			✓	✓	✓											5-Very Hard
HH_C4S4Q03			✓		✓				✓		✓											4-Hard
HH_C3S4Q51	✓					✓			✓	✓								✓				5-Very Hard
HH_C3S4Q53									✓			✓										4-Hard

Fig. 3. Skill-based search of problems.

for the probability that particular student will answer that particular problem correctly.

Fig. 3 shows an example of the search feature where the student has searched for questions requiring the Derivatives Shortcuts skill (DeS) and has eliminated those questions requiring the Chain Rule, Quotient Rule, or Trig skills. The search results in a set of problems that can be ordered by the probability that the system calculates for the likelihood the student will correctly answer the question. It also shows all of the required skills for that question. The student can then click on the name of any question in the list to try it on the spot.

VIII. Using Skills to Provide Value for a Teacher

While the focus of our recent efforts have been on facilitating student learning there are practical benefits for teachers as well. The dashboard allows teachers to track where all the students are in their class. This can be done to whatever group makes sense to the teacher. For example, it might be relative to other members of the current class or to previous years. Thus a teacher can see not only which students are falling behind compared to each other but also see if the class overall is behind where other classes were at a comparable point in previous years.

This information is broken down by skills so the teacher can see in what areas the students in their class need work. If the entire class is behind in specific areas they could tailor a quiz to target the problem areas, finding questions using a method similar to what was described in the student self quiz above. Alternatively custom quizzes could be built for groups of students or for individual students.

IX. Conclusions and Future Work

Our work with PFA provides a way to organize problems by the skills that they require and provides an individualized difficulty based on student history. This information can be used to pinpoint the skills in which a student is falling behind and then facilitate intervention.

While we have a large data sample to show that the question difficulty is meaningful we only have a small amount of qualitative feedback about the dashboard and the personalized searchable question library. Our next steps include putting the dashboard concept to the test by using the system in a full Calculus course. We plan to test both qualitatively and quantitatively whether the use of the CalcTutor system results in increased student learning, increased student satisfaction, and/or more efficient or effective identification of students in need of intervention by the teacher. We expect that results such as these may be seen since the dashboard information would be providing feedback (to both the teacher and students) highlighting particular skills in which students are deficient.

The previous version of the CalcTutor also employed a concept called the "Teacher's Dilemma"[5]. The Teacher's Dilemma is a peer learning framework in which students are rewarded for asking each other "good" questions. Here, "good" is loosely defined as hard questions the other student gets right or easy questions they get wrong. A key part of that is knowing the difficulty of questions. In earlier implementations of software the difficulties that were used were effectively hard coded and very difficult to update. In addition students were given very little information upon which to base their decision other than the difficulty of the question. It is our hope that with a better difficulty metric, question skills, and features like the dashboard we can improve on previous results using that technique.

Finally it should be noted that while the CalctTutor system currently only provides support for an introductory Calculus class there is nothing conceptual preventing updating the framework for other domains. While it would require the work to write questions, create skills, assign skills to questions, etc., PFA is a general framework and could be used in a wide variety of domains. Similarly the CalcTutor question system would require some expansion but has no fundamental limitations preventing it from being applied to other areas.

REFERENCES

[1] K. Kime, R. Torrey, and T. Hickey, "CalcTutor: Applying the teachers dilemma methodology to calculus pedagogy," in *2015 IEEE Frontiers in Education Conference (FIE)*, Oct. 2015, pp. 1–8.

[2] K. Kime, T. Hickey, and R. Torrey, "The calculus dashboard - leveraging intelligent tutor techniques to provide automated fine-grained student assessment," in *2017 IEEE Frontiers in Education Conference (FIE)*, Oct. 2017, pp. 1–8.

[3] P. I. Pavlik, H. Cen, and K. R. Koedinger, "Performance factors analysis–a new alternative to knowledge tracing," in *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*. IOS Press, 2009, pp. 531–538.

[4] M. Gage, A. Pizer, and V. Roth, "WeBWorK: Generating, delivering, and checking math homework via the Internet." in *Proceedings of the 2nd International Conference on the Teaching of Mathematics*, Crete, Greece, 2002.

[5] A. Bader-Natal, *The Teacher's Dilemma: A Game-based Approach for Motivating Appropriate Challenge Among Peers*. ProQuest, 2008.

[6] F. M. Lord, "The Relation of Test Score to the Trait Underlying the Test," *ETS Research Bulletin Series*, vol. 1952, no. 2, pp. 517–549, Dec. 1952.

[7] G. Rasch, *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests*. Oxford, England: Nielsen & Lydiche, 1960, vol. xiii.

[8] H. Cen, K. Koedinger, and B. Junker, "Learning factors analysis a general method for cognitive model evaluation and improvement," in *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, ser. ITS'06. Springer-Verlag, pp. 164–175.