Policy Building – An Extension To User Modeling

Michael V. Yudelson
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213, USA
yudelson@cmu.edu

Emma Brunskill
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213, USA
ebrun@cs.cmu.edu

ABSTRACT

In this paper we combine a logistic regression student model with an exercise selection procedure. As opposed to the body of prior work on strategies for selecting practice opportunities, we are working on an assumption of a finite amount of opportunities to teach the student. Our goal is to prescribe activities that would maximize the amount learned as evaluated by expected post-test success. We evaluate the proposed approach using an existing dataset where data was collected performing random skill selection. Our results cautiously support the hypothesis that using policies designed to optimize the post-test score associated with higher learning outcomes, but more work is needed.

1. INTRODUCTION

Recently there has been significant interest in logistic-regression based student modeling methods, including Performance Factors Analysis [3], Instructional Factors Model [2], and Contextual Factors Analysis [4]. Such models can flexibly incorporate skill difficulties and individualized student parameters. There is evidence that such models outperform Knowledge Tracing in terms of predicting student performance [2]. However, to our knowledge there has been no work that uses such student models with instructional decision making about what skills students should practice or what activity to perform next to maximize learning.

For example, consider selecting between the following problems when teaching a student least common multiples:

1 (Product). Sally visits her grandfather every 2 days and Molly visits him every 7 days. If they are visiting him together today, in how many days will they visit together again?

2 (LCM). Sally visits her grandfather every 4 days and Molly visits him every 6 days. If they are visiting him together today, in how many days will they visit together again?

Problem 1 can be solved by simply multiplying the given numbers (hence the tag *Product*). Problem 2 is an *LCM* and multiplication will not work. An open question is which problem type should be selected, and at what point in the student's learning progress. The seemingly obvious approach of presenting the easier *Product* problem earlier, and the harder *LCM* later on may not be best as emphasis on the use of a partial strategy of solving problems on least common multiples could lead to learning misconceptions. However, starting with harder *LCM* problems too early could be too challenging and might delay learning. In addition, it is likely that which activity to choose should depend on the student's current understanding and student ability.

In this paper we consider automatically selecting among such problems based on an online estimate of the student's probability of getting these problems correct. Our work differs from work on strategies for selecting practice opportunities (or more generally, pedagogical activities) to help the student reach mastery. Instead in our work we assume that the objective is to select a fixed number of activities to give to the student in order to maximize the amount learned, as evaluated by expected post-test success. This may be a useful objective in some classroom settings where a fixed amount of time is available.

One important challenge when considering new methods for problem selection is how to evaluate these methods. Typically student tutoring data is collected using a fixed policy for selecting problems, and if the proposed new policy differs from the prior policy, it can be hard to evaluate it using the prior dataset. In this work we leverage an existing dataset where part of the data was collected by performing random skill selection. This allows us to evaluate the policies we compute by finding existing examples in the dataset that happen to match the proposed policy. We can then compare the empirical performance of the matching examples to the performance of the students' whose policy did not match the proposed policy. In this way we can use existing randomized data to perform a *post-hoc* analysis of alternate policy strategies that can be used.

Though the size of our data prevents any strong conclusions, our preliminary results are promising. They suggest that selecting policies designed to optimize the post-test score are associated with higher post-test scores than other policies. Further work is required to examine this in more detail.

2. APPROACH

We now describe how we model student learning, and then describe how we use these models to create adaptive policies for what activity to select.

2.1 Student Modeling

We use the Contextual Factors Analysis (CFA) [4] framework to model student learning. CFA is an educational data mining model. It was developed as an elaboration on a series of other cognitive models, namely Performance Factors

Algorithm 1 BestNextSkill

```
Input: for student $i$, no. successes on skill 1 and 2, $s_{i1}, s_{i2}$, and no. failures $f_{i1}, f_{i2}$, all other parameters $\psi_i$, no. problems given $d$, no. problems left to give $D$
Output: expected post-test score $score$ for best skill to practice, $bskill$
if $d = D$ then
    $score$=0;
    for $j = 1 : 2$ do
        {predict post-test score}
        $score = score + p(c|skill_j)$
        $skill = NULL$ {No more time to practice}
    end for
else
    for $j = 1 : 2$ do
        $f'_{ij} = f_{ij} + 1$ {practiced skill $j$, failed}
        $fscore_j^{fail}$=BestNextSkill(for $k \neq j : \langle s_{ik}, f_{ik} \rangle$, $s_{ij}$,$f'_{ij}$,$\psi_i$,d+1,D)
        $s'_{ij} = s_{ij} + 1$ {practiced skill $j$, success}
        $fscore_j^{suc}$=BestNextSkill(for $k \neq j : \langle s_{ik}, f_{ik} \rangle$, $s'_{ij}$,$f_{ij}$,$\psi_i$,d+1,D)
        $score(j) = p(f|s_{i1}, s_{i2}, f_{i1}, f_{i2}, \psi_i) * fscore_j^{fail} + p(s|s_{i1}, s_{i2}, f_{i1}, f_{i2}, \psi_i) * fscore_j^{suc}$
    end for
    $score = \max_j score(j)$
    $bskill = \arg\max_j score(j)$
end if
```

Analysis model [3], Additive Factors Model (AFM) [1], and Rasch 1PL IRT model [6]. In addition to account for the number of correct and incorrect attempts to apply a skill separately (as PFA does in contrast to AFM), it captures transfer effects of prior attempts with one skill on the other. A logistic regression form of CFA is given in Equation (1):

$$logit(p_{ij}) = \theta_i + \sum_{a \in Q_j} (\beta_a + \gamma_a s_{ia} + \rho_a f_{ia}) + \sum_{b \notin Q_j} (\gamma_b s_{ib} + \rho_b f_{ib}) \quad (1)$$

Here, p_{ij} is probability that student i solves problem j correctly, θ_i is student's ability parameter, and Q is a so-called Q-matrix [5] that encodes what skills are associated with j^{th} problem (or a problem step). β_a, γ_a, and ρ_a are complexity, success learning rate, and failure learning rate respectively; they pertain to the skill(s) that is (are) addressed in j^{th} problem (or problem step). γ_b, and ρ_b are success and failure transfer rates respectively; they capture transfer from skill b to skill a. s_{ix} and f_{ix} are the number of prior success and failures with x^{th} skill. In our prior work with CFA (rf. [4]) we found it to be superior to PFA, whether or not the transfer parameters (γ_b and ρ_b) were significant. It is due to these reasons that we used CFA.

2.2 Adaptive Instructional Policies

We now consider how to use our student model to automatically create adaptive instructional policies. Consider the scenario where we have 2 different skills we would like the student to learn, and we have a fixed number of opportunities D when we can give the student practice on either skill. We assume as input we are provided the CFA student learning parameters. The objective is to compute an adaptive policy for D skill opportunities should be provided to the student in order to maximize his expected post-test performance on 1 question per skill. The policy computed is an adaptive, conditional policy, because it depends on the responses made by the student: as the student responds to each practice opportunity, we update the number of success and failures of the student over each skill. This in turn will change what is the next best skill practice opportunity to give to the student. The way we compute the policy can be thought of as constructing a forward search tree, where we alternatively consider all possible skill practice opportunities to provide next, and then the possible responses (success or failure) of the student. We repeat this expansion for the desired number of D practice opportunities. At the end of this, at a tree leaf, we compute the expected post-test performance, given the successes and failures of the tree path to this leaf. This simply involves predicting the probability that the student will get a question about skill 1 correct plus the probability they will get a question about skill 2 correct. Both these quantities can be computed using the student model. We repeatedly take expectations and maximizations to use these leaf scores to decide what skill should be practiced at the current student state: see Algorithm 1 for details.[1] Two-steps of a sample adaptive policy are shown in Figure 2.1.

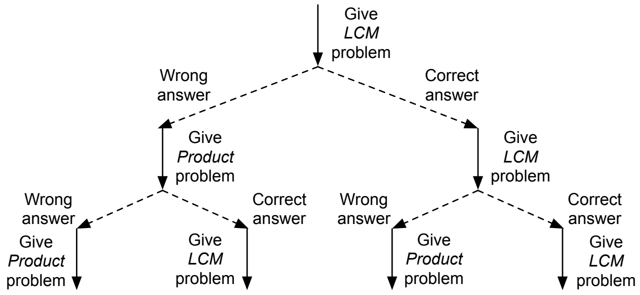


Figure 1: Example adaptive instructional policy.

Note that the computed "optimal" policy that is expected to maximize the student's post test performance is a direct function of the input student parameters. Therefore, the optimal policy can be different for different students.

3. DATA

The data comes from an experimental study conducted at Pinecrest Academy Charter Middle School. Students from 6^{th} and 7^{th} grades were exposed to a modified Carnegie Learning Bridge to Algebra (BTA) tutor. The part of the experiment we analyzed consisted of 10 sessions. In each of the sessions students were given 16 problems randomly drawn from a pool of 24 without replacement. One of the experimental conditions only included 8 problems to be delivered and it was removed for the sake of uniformity. Each session addressed a separate topic. Within a topic there were two or four skills, and the problems covered one or two of them. For example, one session was on least common multiples, and the skills were divided by: 1) whether the problem was formulated as a story or not ("story" or "word" problems), and 2) whether a solution can be obtained by mere multiplication or not ("product" and "true least common

[1] An alternative, but equivalent, method is to have Algorithm 1 return a complete conditional policy tree showing what skill to give after each possible student response.

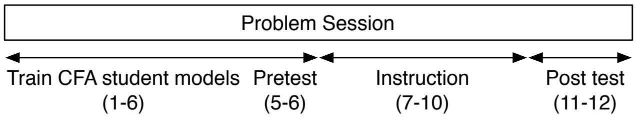


Figure 2: A topic session of 12 problems was divided into sections that we used to fit student models and consider pre and post performance after a period of 4 problems.

multiple" problems). In our analysis we group problems so that we considered only 2 alternate skills at a time.

4. EXPERIMENT

To evaluate our approach, we segmented each student's session data as follows (cf. Figure 3). Problems 1-6 were used to train the CFA models. These models were used to compute the instructional policy for a student to maximize expected post test score after doing 4 problems. The student's performance on problems 5-6 were used as a pretest score, then problems 7-10 were considered the tutoring/instructional phase, and the student's performance on problems 11-12 were considered a post test. Recall that the problems were selected randomly in the dataset that we used. We only used the first 12 problems (with a 4 problem "instructional" period) so that we could increase the likelihood of finding some overlaps in the data with the computed optimal 4-problem adaptive policies. Therefore we selected the subset of students who happened to get 1 problem for each of the 2 skills we considered in both the pretest and the post-test.

For comparison we also considered two alternate policies. One policy is to always give the student a problem for the skill that the student is more likely to solve correctly. We will call this policy an "easier problem" policy or just an "easy" policy. Our second comparison policy is to always provide the student with a problem that is for the skill that the student is less likely to solve correctly. We will call this policy a "harder problem" instructional policy or a "hard" policy. This harder policy is very similar to a common instructional approach used in Knowledge Tracing mastery learning in which a student is given an exercise for a skill that the student is least likely to have mastered.

We will compare the learning gains of students whose provided problems happened to match the 3 policies of interest (optimal, easy, or hard).

5. RESULTS

Data restriction. We focused our attention on the subset sessions where students improved between the pre- and post-test trials. The summary of learning effects between pre- and post-test trials is given in Table 1. Some sessions are listed twice (sessions 1, 3, and 6) because they contained multiple skills that will be divided into groups (e.g. Story-Word vs. Product-LCM in session 1). Sessions 5, 8, 9, and 10 were not considered because they contained errors in the data. We excluded sessions 2, 3 (both 3.1 and 3.2 versions), and 6.1 because students did not make measurable learning gains.

Policy Performance. A summary of the results of computing optimal policies for the students is given in Table 2. Recall that we compute an optimal policy for each student based on their student parameters. We then find instances in the data where the provided problems happened to match the optimal policy we computed. We repeat this process with the easy policy and the hard policy. Note that it is quite unlikely that the randomly selected problems will happen to match any of the 3 policies. Therefore it is not surprising that the number of matches we find in the data for each of the 3 policies is quite low, ranging from 1 to 14 for optimal policies and from 0 to 7 for comparison policies. Table 2 also lists number of students that follow overlaps of optimal and ad hoc policies.

Table 1: Learning between pre-test (trials 5 and 6) and post-test (trials 11 and 12)

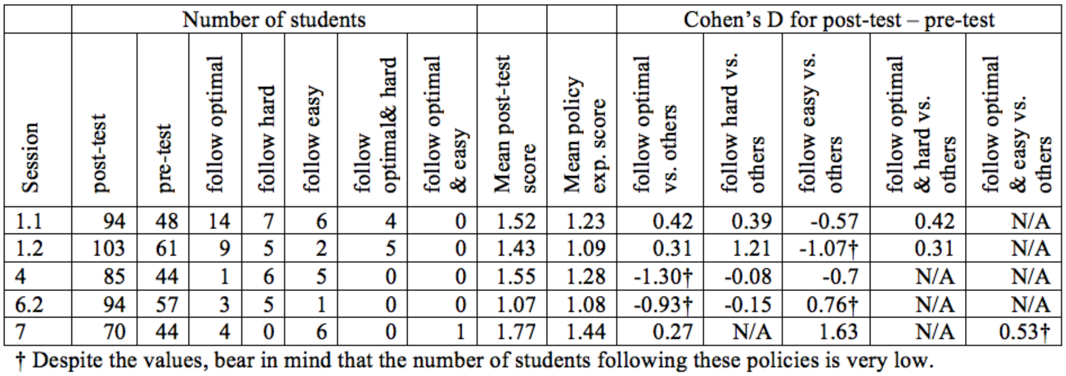
Session	No. stud.	Mean pre-test score	Mean post-test score	Learn. effect size	Learning t-test p-val
1.1	48	1.06	1.52	0.73	0.000***
1.2	61	1.10	1.43	0.44	0.004**
2	51	1.76	1.80	0.09	0.299
3.1	60	0.93	0.93	0.00	0.500
3.2	47	1.00	1.09	0.11	0.280
4	44	1.23	1.55	0.45	0.009**
6.1	53	0.98	1.21	0.29	0.038*
6.2	57	0.86	1.07	0.29	0.035*
7	44	1.41	1.77	0.59	0.002**

The last 5 columns of Table 2 show the comparison between students that received a particular policy versus all other students. Though we caution against making sweeping claims because the number of students that followed any of the policies is very low, there remain some encouraging results. First, for session 1.1 and 1.2, students that received the optimal policy did better than than students that did not. The results were not significant, but trending that way (paired t-test p-value=0.090). In the other 3 sessions it is extremely difficult to assess any trends, as there were very few students that followed any policy at all.

It is not yet clear if optimal policies are significantly better than the comparison policies. In session 1.2 9 matches to the optimal policy are on average only 0.31 standard deviations apart from the rest, while the 5 matches to *hard* policy are more than 1 standard deviation different from others. Interestingly, here matches of the *hard* ad hoc policy are a subset of those who received the optimal policy. It may be that those who received the *hard* ad hoc policy that drive most of the distinctive power of optimal policy. In session 1.1, 7 recipients of the *hard* ad hoc policy are a subset of followers of optimal policy as well. In both session 1.1 and session 1.2, receiving a harder item at every step during a period of interest seems to be universally beneficial with respect to post-test result. In contrast, in session 7, where complying or not with the *easy* ad hoc policy distinguishes students far better than optimal policy. Here, an easier problem at each of the trials of interest is more beneficial. Note that in general the optimal policy is just aiming to maximize the expected student post test performance, and it may not outperform other policies in particular individual cases.

Qualitative Assessment. We also wished to further assess the resulting optimal instructional policies, using insight from the student model parameters. Table 3 shows the CFA model parameters that were fit using all 16 prob-

Table 2: Summary of student policy data.



	Number of students									Cohen's D for post-test – pre-test				
Session	post-test	pre-test	follow optimal	follow hard	follow easy	follow optimal& hard	follow optimal & easy	Mean post-test score	Mean policy exp. score	follow optimal vs. others	follow hard vs. others	follow easy vs. others	follow optimal & hard vs. others	follow optimal & easy vs. others
1.1	94	48	14	7	6	4	0	1.52	1.23	0.42	0.39	-0.57	0.42	N/A
1.2	103	61	9	5	2	5	0	1.43	1.09	0.31	1.21	-1.07†	0.31	N/A
4	85	44	1	6	5	0	0	1.55	1.28	-1.30†	-0.08	-0.7	N/A	N/A
6.2	94	57	3	5	1	0	0	1.07	1.08	-0.93†	-0.15	0.76†	N/A	N/A
7	70	44	4	0	6	0	1	1.77	1.44	0.27	N/A	1.63	N/A	0.53†

† Despite the values, bear in mind that the number of students following these policies is very low.

Table 3: Session 1, *Product* problems vs. *LCM* problems. User modeling parameters of recessed (CFA$_{1-6}$) and full (CFA$_{1-16}$) models with respective p-values

Parameter	CFA$_{1-6}$	CFA$_{1-16}$
bias	-1.558(0.000***)	-0.824(0.000***)
$\beta_{Product}$	1.575(0.000***)	1.143(0.000***)
$\gamma_{Product}$	0.109(0.482)	0.124(0.020*)
$\rho_{Product}$	0.861(0.000***)	0.219(0.002**)
γ_{LCM}	0.155(0.235)	0.389(0.000***)
ρ_{LCM}	0.397(0.000***)	0.080(0.019*)
$\gamma_{Product \rightarrow LCM}$	0.071(0.563)	-0.003(0.948)
$\rho_{Product \rightarrow LCM}$	0.554(0.000***)	0.032(0.582)
$\gamma_{LCM \rightarrow Product}$	-0.272(0.087.)	0.094(0.036*)
$\rho_{LCM \rightarrow Product}$	0.209(0.067.)	0.089(0.021*)

lems in a session focused on teaching least common multiples. This model (CFA$_{1-16}$) has parameters that indicate learning from successes and failures for both *LCM* and *Product* problems. Transfer learning is significant and positive from a harder *LCM* to an easier *Product* problem, but the reverse direction (from *Product* to *LCM*) does not show significant transfer. This suggests that *LCM* problems help the student improve on both *LCM* and *Product* problems, but *Product* problems only produce improvement on *LCM* problems. Further this suggests that during tutoring it is likely to be more beneficial to provide *LCM* problems than *Product* problems.

For the LCM topic there were 14 out of 94 students that followed their respective optimal policies. The paths that these students took during trials 7 through 10 consisted of *LCM* problems only. This matches what we might expect given the CFA$_{1-16}$ model that demonstrates the particular transfer benefit of *LCM* problems. None of the paths of other 80 students were composed of solely *LCM* problems.

6. DISCUSSION

It is too preliminary to draw any definitive conclusions from this work because of the limitations of our dataset. From about 200-250 students in each session we had to select a subset that met our criteria of receiving different problem items on pre- and post- test trials. As a result the numbers shrunk to 70-100 students. Within this restricted set the student recipients of the 3 policies were very few.

There needs to be further work to better understand if simple policies are equally effective to the optimal policies. In this dataset we saw several instances of this. However, this could be due to fitting CFA models on a small data set covering only a few hundred students. It also could be because there was only a very small number of students where the problems selected matched any of the considered policies.

As part of the future work, we would like to repeat described experiments on several other datasets, potentially from different subject domains, where randomized data is available. Should the results turn out to continue support the preliminary evidence that optimized policies lead to better post-test performance, we would like to design an experiment using these policies to select skill practice for students.

7. ACKNOWLEDGEMENTS

This research was made possible with the assistance and funding of the U.S. Department of Education (IES-NCSER award #R305B070487), Carnegie Learning Inc., the Pittsburgh Science of Learning Center, DataShop team (NSF-SBE award #0354420) and Ronald Zdrojkowski.

8. REFERENCES

[1] H. Cen, K. Koedinger, and B. Junker. Comparing two IRT models for conjunctive skills. *ITS* 2008.

[2] M. Chi, K. Koedinger, G. Gordon, P. Jordan, and K. VanLehn. Instructional factors analysis: A cognitive model for multiple instructional interventions. *EDM* 2011.

[3] P. Pavlik, H. Cen, and K. Koedinger. Performance factors analysis - a new alternative to knowledge tracing. *AIED* 2009.

[4] P. Pavlik, M. Yudelson, and K. Koedinger. Using contextual factors analysis to explain transfer of least common multiple skills. *AIED* 2011.

[5] K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354, 1983.

[6] W. van der Linden and R. Hambleton (Eds.), *Handbook of Modern Item Response Theory.* Springer, New York, NY, 1997.