Toward Near Zero-Parameter Prediction Using a Computational Model of Student Learning

Daniel Weitekamp III
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
weitekamp@cmu.edu

Erik Harpstead
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
harpstead@cmu.edu

Christopher J. MacLellan
Soar Technology, Inc.
3600 Green Court, Suite 600
Ann Arbor, MI 48105
chris.maclellan@soartech.com

Napol Rachatasumrit
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
napol@cmu.edu

Kenneth R. Koedinger
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
koedinger@cmu.edu

ABSTRACT

Computational models of learning can be powerful tools to test educational technologies, automate the authoring of instructional software, and advance theories of learning. These mechanistic models of learning, which instantiate computational theories of the learning process, are capable of making predictions about learners' performance in instructional technologies given only the technology itself without fitting any parameters to existing learners' data. While these so call "zero-parameter" models have been successful in modeling student learning in intelligent tutoring systems they still show systematic deviation from human learning performance. One deviation stems from the computational models' lack of prior knowledge—all models start off as a blank slate—leading to substantial differences in performance at the first practice opportunity. In this paper, we explore three different strategies for accounting for prior knowledge within computational models of learning and the effect of these strategies on the predictive accuracy of these models.

1. INTRODUCTION

A computational theory approach to modeling psychological phenomenon consist of building simulations of cognitive processes and testing their ability to emulate and explain human behavior. Within educational data mining there have been several attempts to model student learning processes using a computational theory approach. For example SimStudent [1] is a computational approach which simulates human students. The Apprentice Learner (AL) Architecture [2] is a modular framework for creating computational agents such as SimStudent that learn to solve problems in intelligent tutoring systems (ITSs) as a human student would. These computational agents mimic the inductive learning process undergone by students in response to examples and correctness feedback given in a particular domain.

These approaches afford several different use cases relevant to the domain of EDM. First, as cognitive models of human learning, these computational agent models can be used to test theories of human learning. The AL framework uniquely serves this role since it is designed so that its components can be interchanged, to enable the instantiation of different theories of learning that can be tested against human behavior.

Second, insofar as as computational agents constitute high fidelity models of human learning as simulated students, they can be used as cognitive crash dummies for instructional design prior to the longer process of classroom trials and A/B studies. For example, experiments with SimStudent and agents built with AL showed that interleaving fraction addition and fraction multiplication problems would lead to more efficient learning by opportunity in an ITS [14]. This result was later corroborated by human trials [15].

Finally, simulated students can be used as efficient authoring tools. Simulated students can form generalized production rules from examples and correctness feedback meaning that they can be trained in a manner similar to example tracing [3] to build expert models for ITSs. Expert models trained with SimStudent, for example, have similar expressivity and accuracy as hand-written production rule models, but can be built in significantly less time and without any programming knowledge required by the author [12, 11, 16].

Computational theory approaches of learning such as SimStudent and the simulated students built with the AL framework are at their core zero-parameter models of human learning. These approaches are distinct from performance pattern models such as the Additive Factors Model (AFM) [4, 5] or Bayesian Knowledge Tracing (BKT) [6] since they attempt to predict the presence of a pattern in behavior (e.g., learning a new skill or misconception) given only the task environment and a few underlying assumptions about the learning process rather than fitting to data from individual students. These computational theory approaches to human

Daniel Weitekamp, Erik Harpstead, Napol Rachatasumrit, Christopher Maclellan and Kenneth R. Koedinger "Toward Near Zero-Parameter Prediction Using a Computational Model of Student Learning" In: *The 12th International Conference on Educational Data Mining*, Michel Desmarais, Collin F. Lynch, Agathe Merceron, & Roger Nkambou (eds.) 2019, pp. 456 - 461

modeling attempt to make predictions of human behavior without knowing anything about the humans that they are modeling, putting the burden of the modeling process on deliberate and explainable algorithmic design choices. While there are added challenges to designing computational theory approaches that accurately reflect human behavior, they have the advantage of fine-grained explainability. Since a computational theory approach actually performs the tasks given to the students that it attempts to model, the fidelity of the model to its human counterparts can be evaluated in a more incisive manner, considering not just the performance of the model by opportunity, but the types of errors it makes and strategies it employs as well.

While computational theory models have demonstrated some success in the past they still possess several systematic issues in their performance. When applied to the task of modeling learning many computational theory models manifest some version of a cold start problem. Namely, as these models, by definition, are not fit to information from individual students, they often lack the ability to account for individual differences in prior knowledge or experience that students bring to bear. Performance pattern models are free to deal with this problem in a number of ways. For example, AFM and its variants often fit some kind of student intercept to allow for variation among students' initial ability [4, 5]. Several strategies for estimating Bayesian priors in BKT have been proposed in the EDM literature [6, 7]. In a performance pattern context estimating prior knowledge amounts to estimating some parameter between 0 and 1. Computational theory models, on the other hand, require a fully specified mechanistic skill to be initialized and refined in a learning process that directly models the process of a human trying to master skills in an intelligent tutoring system.

In this paper we explore several strategies for accounting for prior knowledge in the AL architecture and compare each strategy's ability to generate learner performance similar to that of students. We close with a discussion of these strategies and discuss the limitations of our current approaches as well as directions for future work.

2. OUR EXPERIMENT

2.1 The Apprentice Learner Architecture

The Apprentice Learner (AL) Architecture is a modular framework for modeling the human learning process from demonstrations and feedback [2]. The purpose of the architecture is to serve as a test-bed for computational theory approaches to student modeling. Agents created in the AL framework induce production rules from training that can be provided either interactively by a human or by working with an existing ITS. In our experiments we use the latter approach to train a set of simulated students on an intelligent tutoring system with the same order of problems as each student from a human dataset.

AL agents receive two types of feedback from ITSs, examples and correctness feedback. When an AL agent has a learned production rule that can fire it will try the resulting action and get correctness feedback from the ITS, otherwise the agent will request a hint (i.e., an example) from the ITS. Through positive examples from hint requests and positive feedback, and negative examples from negative feedback, AL refines its understanding of the domain and learns to correctly solve problems in the ITS.

To induce a production rule model suitable for solving problems, AL employs a four mechanism learning process consisting of *when-learning*, *where-learning*, *how-learning*, and *which-learning*. The *when-* and *where-learning* mechanisms determine the context in which a production rule should fire, the *how-learning* mechanism searches for chains of overly general operators that explain tutor demonstrations to determine what a production should do when it fires, and the *which-learning* determines which production rule should fire if multiple are applicable.

The operators that *how-learning* employs are "overly general", in the sense that they are applicable in a wide range of domains, but not specific enough to solve problems in any particular domain. Overly general operators do not constitute production rules since they are not programmed with any sense of when they should fire. In our experiments we use one perceptual operator 'equals' which adds to the problem state the fact that two values are equal, and four procedural operators 'add', 'subtract', 'multiply' and 'divide'. The conditions in which these operators should be utilized, the particular interface elements from which they derive their numerical inputs, and the interface elements that will receive their outputs are learned by the *when-* and *where-learning* mechanisms respectively to create full production rules.

The AL agents in our experiments employ Trestle [13], an incremental hierarchical categorization algorithm for *when-learning*. For *where-learning*, since we use only static interfaces in our experiments, we employed a 'most specific' strategy, which uses the inputs and outputs present in positive training examples without trying to generalize from those examples to unseen cases. We allow our *how-learning* mechanism to only use single operation explanations (i.e. just multiply or add two numbers, not three numbers or combinations of operators). The *which-learning* mechanism chooses the production rule which has been employed successfully with the highest frequency so far given the current state representation.

2.2 Data Source

To evaluate different methods of pretraining AL agents we used student performance data from an ITS for fraction arithmetic [15]. The dataset consisted of the work of 117 students on three different types of fraction arithmetic problems: 1) adding fractions with the same denominator, 2) adding fractions with different denominators, and 3) multiplying fractions. In the interface students are first asked if they need to convert the two fractions, which is false in the addition-same and multiplication cases and true in the addition-different case. If the fractions need to be converted then the students must find the common denominator between them and write in the converted fractions before adding them. This dataset is available on DataShop [9] [1].

[1]https://pslcdatashop.web.cmu.edu/Project?id=243

2.3 Training Strategies

We explored three different strategies for accounting for prior knowledge in AL Agents. For each of these strategies an agent was paired with a human student and provided some form of pretraining based on data collected about that student prior to working through that student's problem sequence in the data. In our experiments we attempted three different strategies to account for prior knowledge estimated fraction prior experience, estimated whole number prior experience, and demonstrated pretest. Additionally, we ran a no pretraining control condition which begins from a conventional cold start.

2.3.1 Estimating Prior Experience

In the estimated fraction and estimated whole number prior experience strategies agents worked through a number of pretraining problems based on estimates of each student's prior opportunities to practice problems in each of the three types of fraction arithmetic problem. We estimated the number of prior opportunities (P_{ik}) that a student i had on each knowledge component (KC) k [10] in the fraction arithmetic tutor. Taking the AFM mixed model regression equation:

$$log(\tfrac{p_{ij}}{1-p_{ij}}) = \theta_i + \Sigma_k(q_{jk}\beta_k + q_{jk}\gamma_k T_{ik}), \theta_i \sim \mathcal{N}(0,\, \sigma^2)$$

Consider an imaginary tabla-rasa student with no knowledge of anything at all with student intercept $\theta_{-\infty}$, and a student with student-intercept θ. We can find the number of prior opportunities P_{ik} such that the tabla-rasa student has the same log odds of answering a question in KC k as student i:

$$\theta_{-\infty} + \Sigma_k(q_{jk}\beta_k) + \Sigma_k(q_{jk}\gamma_k P_{ik}) = \theta_i + \Sigma_k(q_{jk}\beta_k)$$

$$\implies P_{ik} = \tfrac{\theta_i - \theta_{-\infty}}{\gamma_k}$$

An issue with this formulation is that a true tabla-rasa student would have $\theta_{-\infty}$ equal to -inf, so we introduce an approximation for $\theta_{-\infty}$ in order to get reasonable values for P_{ik}. In our experiments we choose $\theta_{-\infty}$=-2 since this is the student intercept at which a student practicing a single KC step with a KC intercept of zero would have about a 10% probability of getting the step correct. Although this an arbitrary choice, we believe it is a reasonable one, much in the same way that it is reasonable to choose a 90% chance of correct behavior as a mastery threshold in a BKT model.

There are three distinct types of fraction arithmetic problems, which each have their own sets of KCs. In the case of fraction addition the two fractions can either have the same denominator, in which case the student need only add the numerators, or the denominators could be different, requiring the student to convert the fractions before adding. In the third case, fraction multiplication, the KCs are distinct from the addition cases. Since in practice fraction arithmetic problems are not presented with any of their KCs in isolation we estimated the number of problems of each type to give as prior training to our learning agents as the minimum P_{ik} among the problems of a given type. For the estimated fraction prior experience case we pretrain on a number of random problems from each type according to these estimates. In the estimated whole number prior experience case we pretrain as many random whole number addition problems as there are estimated same denominator problems and as many random whole number multiplication problems as estimated multiplication problems.

In the estimated fraction condition, randomly generated problems were restricted to have denominators between 2 and 12 with numerators less than each fraction's denominator (i.e., no improper fractions). In the estimated whole number condition, each agent was given randomly generated whole number arithmetic problems prior to beginning the core tutoring sequence. These whole number arithmetic tutors were restricted to numbers between 1 and 12.

2.3.2 Demonstrating Prior Answers

In the demonstrated pretest case we pretrain the AL agents by providing them with demonstrations of the exact answers that students gave in a pretest evaluation of the original study. In order to model both the knowledge and misconceptions of the student, these answers are given to AL as positive examples regardless of whether or not each of the students' answers were correct. The goal here is that the pretest encapsulates a picture of each students' prior conceptions that may contain more information than a fit parameter. The pretest is a snapshot of students behavior under a particular set of problems, a sample which we use to infuse an associated AL agent with the same knowledge and misconceptions as the original student.

One limitation of our pretest demonstration approach is that the available pretest data only contained the given fraction problems and the students' final answers. Thus, the demonstrations that we provided to the agents appeared as single steps even though the human students may have done multiple mental calculations to arrive at their answer. This issue is likely to be particularly salient on fraction addition problems with different denominators as the common denominator process would not be apparent.

3. RESULTS

In Figure 1 we see that the estimated fraction condition gets closest to the human first opportunity performance, followed by the demonstrated pretest condition. The estimated whole number condition behaves almost equivalent to the control case with a 100% first opportunity error rate.

To test the fit of each strategy to the human data we calculate the residuals between the human learning curves and the learning curves generated from the AL agents run with each pretraining strategy (Figure 2). Table 1 shows several statistics of the fit of each strategy to the human data. The "Accuracy" column shows the mean accuracy between the correctness (0 for incorrect and 1 for correct) of each human and that human's AL agent counterpart over all students and opportunities. The "First Opp. Accuracy" column shows this same statistic, but only for the first opportunity. In both cases the estimated fraction strategy shows the best fit to the human data.

In addition to testing the predictive accuracy of the models,

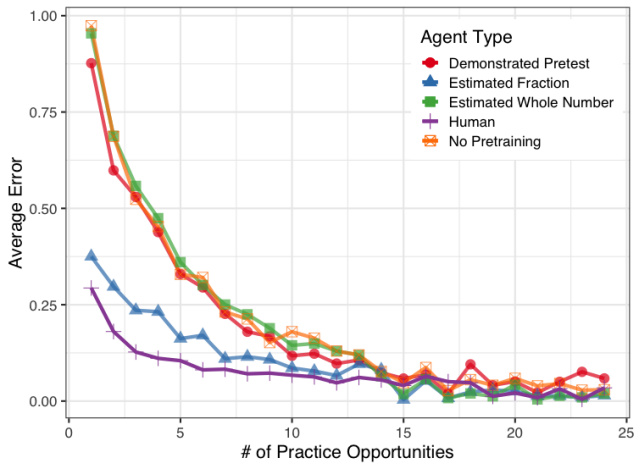

Figure 1: Learning curves aggregated over all knowledge components for each pertaining strategy and the human learning curves.

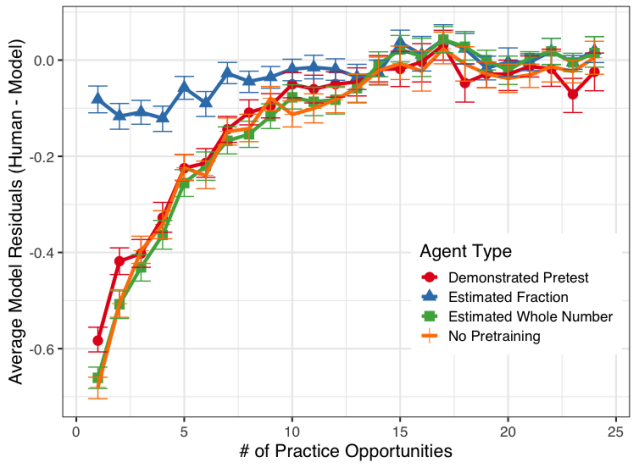

Figure 2: Learning curve residuals (human minus agent) for each strategy across all knowledge components. Error bars are standard errors.

we next looked at the explanatory power of the model– by looking at its ability to account for variation in the human error rates over the course of problem solving. As a coarse measure of explanatory power, we first conducted a χ^2 test of independence between each model correctness and the human correctness. This test confirmed that there is a significant relationship between each models' predictions and the human correctness (independence hypothesis rejected for all models, $p < 0.001$). Next, we looked to more finely evaluate how well each model explains variation in the human data. To do this we used a mixed-effects regression analysis [16]. For this analysis, we used a mixed-effects model with fixed effects for each model prediction as well as the practice opportunity counts. The model also included random effects for the intercept and slopes of each knowledge components as well as intercepts for each student.[2] This model, which is effectively the Additive Factors Model [4, 5] with an additional term for the model predictions, serves a role similar to a repeated measures ANOVA analysis by accounting for general effects of within student and skill (knowledge component) performance as well as the effect of repeated practice. The model enabled us to evaluate how well each computational models' predictions improved the overall regression model fit over the baseline AFM model. By applying model fit statistics, such as AIC, to these mixed-effects models, we could evaluate which model better accounted for variation in the human behavior above general practice effects and general correlations of behavior within students and skills. The baseline AFM model fit to this data had an AIC score of 9558.7. The "AIC" column in Table 1 shows the AIC scores for the mixed-effects model with the added fixed effects for each respective computational model. Note, the numbers of free parameters and data points between each of these models is equivalent thus we did not consider other fit statistics such as BIC or HQC as they would have resulted in an identical ordering.

These results show that all models provide some explanatory power over the baseline AFM model. However, the no pretraining control model appears to provide the most explanatory power. Although this result seems counter-intuitive from the graphs shown in Figure 1, there is at least one possible explanation. Mainly, that the estimated fraction model better fits the average of all students whereas the no pretraining model better fits individual students– specifically students with high initial error. The estimated fraction model likely better fits the students that perform better initially; however these students have less variation in their behavior that the model might account for and this variation is already accounted for as a general within-student effect of the regression evaluation. In contrast, the no pretraining control and demonstrated pretest models better fit students with lower performance at the beginning, who also have more variation to account for.

Finally, we analyzed the residuals shown in Figure 2 using a linear regression analysis. This analysis fit a line to each residual curve to get an estimate of the intercept and slope of these curves. Table 2 shows the slopes and intercepts of these linear models with their accompanying 95% confidence regions. This analysis shows there is a significant intercept and slope for each of the models (all intercepts and slopes are non-zero, $p < 0.001$). However, the estimated fraction model has the intercepts and slopes that are closest to zero, suggesting that it is a better fit of the overall error rates and error rates by opportunity.

Table 1: Fit statistics to human data by strategy.

Strategy	Acc.	First Opp. Acc.	AIC
No Pretraining	0.684	0.316	**9541.3**
Est. Fractions	**0.805**	**0.643**	9558.1
Demo Pretest	0.701	0.389	9548.2
Est. Whole Num	0.681	0.326	9556.4

[2]The mixed-effect regression model used in R: Human.Correctness ~ Agent.Correctness + Opportunity + (1 + Opportunity | KnowledgeComponent) + (1 | Student.Id).

4. DISCUSSION

Our current experiments in accounting for prior knowledge in AL agents have mixed results, but yield fruitful directions for future work. Our most promising method, the estimated fraction prior experience case, reduces the average first opportunity error rate across all knowledge components down to about 40%. The human students by contrast begin at about a 30% error rate. The demonstrated pretest case yielded little improvement over the control. The only slight first opportunity improvement in the demonstrated pretest case is likely due to the often erroneous answers that the students produced in the pretest, which we presented to our AL agents as the ground truth in this condition. Our original hypothesis was that these erroneous but positively labelled examples would account for any existing misconceptions in the human students. However, these erroneous examples may have gone too far toward confusing the AL agents. An alternate explanation is that the lack of negative examples in this case hinders the AL agents' capacity to effectively delineate between their learned production rules. Finally, since the estimated whole number learning curves turned out to behave equivalently to the control case both in first opportunity error rate and in over-all shape, we need to look more closely at AL's capabilities for cross-interface skill transfer in future work .

We consider these experiments to be a first exploratory pass at accounting for prior knowledge in computational modeling approaches to human learning. Among them, the estimated fraction prior experience case is decidedly the closest to human behavior. However, we certainly think we can get much closer to the human behavior going forward.

One direction for future work is to improve our estimates for the number of opportunities to pre-train our agents with to account for the prior knowledge of their human counterparts so that the intercepts of the human and AL agent learning curves match. We can approximate the average of the ground truth number of prior opportunities by calculating the number of opportunities that the control condition takes to gain parity with the human students at their first opportunity. By comparison to this ground truth average it appears that our estimates are too large in the multiplication (14.17 vs. 6) and same denominator addition problems (6.76 vs. 4) and too low in the addition different problems (6.79 vs. 9). We found that these discrepancies could not be remedied by a different choice of $\theta_{-\infty}$.

In previous evaluations of the AL framework in other domains AL agents learned more per opportunity than the human subjects [16]. We hope to explore this fact further by evaluating situations where AL agents over/under performs relative to the humans, and find ways of correcting the discrepancy.

Table 2: Linear model fit to residuals.

Strategy	Intercept	Slope
No Pretraining	-0.474 ± 0.012	0.030 ± 0.002
Est. Fractions	$\mathbf{-0.108} \pm 0.011$	$\mathbf{0.007} \pm 0.002$
Demo Pretest	-0.422 ± 0.012	0.026 ± 0.001
Est. Whole Num	-0.490 ± 0.012	0.031 ± 0.001

One direction for future work that will help us converge on an accurate model of a human learning, is to give AL agents the capability to make random guesses guided by statistics of human errors. This behavior might include inputting common random numbers, copying random numbers, and applying overly general operators at random. Much in the same way that we have estimated the P_{ik} from human data, we would need to estimate a distribution from which these behaviors could be drawn. Similar statistics could be estimated for inferential and procedural slip mistakes [3].

The question certainly is raised: if we must rely on descriptors of human behavior derived from fitting statistical models to human performance, then to what extent are we employing a zero-parameter approach to modeling? No doubt, we are using fit parameters in our modeling approach by bootstrapping our models guided by student intercepts calculated from AFM. However, our computational modelling approach attempts to replicate human behavior instead of only fitting performance, and this process is guided a priori via computational theories of student learning, not by comparison to human data.

Although our nearly zero-parameter computational approach opens up many possibilities for testing theories of human learning, it comes with its own set of difficulties and limitations. In contrast to performance estimation models such as AFM and BKT which require only the logs of human performance on an ITS, our approach also requires a working ITS for the AL agent to work against. This limits the applicability of our method to older datasets for which the actual tutoring interfaces have been lost to time. Currently our system works on the newest HTML version of CTAT and has been used successfully in previous experiments on the older Java version of CTAT as well. However, in order to use different ITSs new code must be written to communicate tutor events to the AL framework's RESTful API.

While a nearly zero-parameter computational approach to human learning allows us to form and test theories of learning in a very detailed manner, it should be noted that the strength of any claim about the underlying cognitive processes of students is dependant on both the specificity of such a model in replicating human behavior and the generalizability of such a model across different domains. Here we have looked at just a single domain, fraction arithmetic, since it is amenable to all three of our proposed strategies. However, it should be noted that although validity claims are strengthened in a zero-parameter computational approach by the fact that behavior must be explained on an algorithmic level and not by fitting parameters, it is still possible that two underlying learning mechanisms yield almost indistinguishable behavior. Thus generalizability is essential to any claim about human cognitive processes. However, generalizability cannot be gained for free by the availability of more diverse data, as is often the case with deep learning models. Rather, any observed discrepancies between domains must be explicitly explained and accounted for by the investigator at an algorithmic level. This fact lends very high explainability to a zero-parameter computational approach, but nonetheless presents an added challenge.

[3]These statistics would be difficult to estimate convincingly with a zero-parameter approach.

5. CONCLUSION

We have tested three different strategies for accounting for prior knowledge in AL agents trained on fraction arithmetic problems. Our three strategies were 1) 'estimated fraction prior experience' where we gave the AL agents additional practice in fraction arithmetic problems before starting the core tutor problems, 2) 'demonstrated pretest' where we showed the AL agents their human counterparts' answers on pretest problems before starting the core tutor problems, and 3) 'estimated whole number prior experience' where we pretrained the AL agents with whole number arithmetic problems. Our results showed that in terms of matching human learning curves the estimated fraction case beats out all the other strategies. However the control case, which had no pretraining at all, best explains the variance in the data.

We have discussed several limitations of our nearly zero-parameter computational approach to testing theories of learning, and have offered several avenues for improvement. Concerning the limited accuracy of our current approach in estimating the number of prior opportunities necessary for an AL agent to account for the prior knowledge of its human counterpart, we have offered an avenue for further refinement. Additionally we have discussed ways that we might accounting for non-deterministic human behavior such as initial guessing. Finally, we have addressed the zero-parameter nature of our approach and considered its technical and epistemological limitations and strengths. We consider the AL framework to be a strong avenue for testing theories of learning, and hope to refine our computational approach to modeling human learning in future work.

6. ACKNOWLEDGEMENTS

The research reported here was supported in part by a training grant from the Institute of Education Sciences (R305B150008). Opinions expressed do not represent the views of the U.S. Department of Education.

7. REFERENCES

[1] Matsuda, N., Cohen, W. W., Sewall, J., Lacerda, G., & Koedinger, K. R. (2007). Predicting students' performance with simstudent: Learning cognitive skills from observation. Frontiers in Artificial Intelligence and Applications, 158, 467.

[2] Maclellan, C. J., Harpstead, E., Patel, R., & Koedinger, K. R. (2016, June). The Apprentice Learner architecture: Closing the loop between learning theory and educational data. In EDM (pp. 151-158).

[3] Aleven, V., McLaren, B., Sewall, J., & Koedinger, K. R. (2009). Example-tracing tutors: A new paradigm for intelligent tutoring systems. Int J of AI in Education, 19, 105-154.

[4] Cen, H., Koedinger, K. R., & Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. In M. Ikeda, K.D. Ashley, T.- W. Chan (Eds.) Proceedings of the 8th International Conference on Intelligent Tutoring Systems, 164-175. Berlin: Springer-Verlag.

[5] Pavlik Jr., P.I., Cen, H., Koedinger, K.R.: Learning Factors Transfer Analysis: Using Learning Curve Analysis to Automatically Generate Domain Models. In: Barnes, T., Desmarais, M., Romero, C., Ventura, S. (eds.) Proceedings of the the 2nd International Conference on Educational Data Mining, Cordoba, Spain, pp. 121-130 (2009).

[6] Baker, R. S., Corbett, A. T., & Aleven, V. (2008, June). More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In International conference on intelligent tutoring systems (pp. 406-415). Springer, Berlin, Heidelberg.

[7] Hawkins, W. J., Heffernan, N. T., & Baker, R. S. (2014, June). Learning Bayesian knowledge tracing parameters with a knowledge heuristic and empirical probabilities. In International Conference on Intelligent Tutoring Systems (pp. 150-155). Springer, Cham.

[8] Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2006, June). The cognitive tutor authoring tools (CTAT): preliminary evaluation of efficiency gains. In International Conference on Intelligent Tutoring Systems (pp. 61-70). Springer, Berlin, Heidelberg.

[9] Koedinger, K. R., Baker, R. S., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A data repository for the EDM community: The PSLC DataShop. Handbook of educational data mining, 43, 43-56.

[10] Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2012). The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. Cognitive Science, 36(5), 757âĂŞ798. https://doi.org/10.1111/j.1551-6709.2012.01245.x

[11] MacLellan, Koedinger & Matsuda (2014). Authoring Tutors with SimStudent: An Evaluation of Efficiency and Model Quality. Proc of Intelligent Tutoring Systems, 551-560.

[12] Matsuda, Cohen, Koedinger (2015). Teaching the teacher. Int J of AI in Ed, 25, 1-34.

[13] MacLellan, C. J., Harpstead, E., Aleven, V., & Koedinger, K. R. (2016). Trestle: a model of concept formation in structured domains. Advances in Cognitive Systems, 4, 131-150.

[14] Li, N., Cohen, W. W., & Koedinger, K. R. (2012). Problem Order Implications for Learning Transfer. In Proceedings of Intelligent Tutoring Systems, 185-194.

[15] Patel, R., Liu, R., & Koedinger, K. R. (2016). When to Block versus Interleave Practice? Evidence Against Teaching Fraction Addition before Fraction Multiplication. In Proceedings of the 38th annual meeting of the cognitive science society. Philadelphia, PA.

[16] MacLellan, C. (2017). Computational Models of Human Learning: Applications for Tutor Development, Behavior Prediction, and Theory Testing (Unpublished doctoral dissertation). Carnegie Mellon University, Pittsburgh, Pennsylvania