How Well do Contemporary Knowledge Tracing Algorithms Predict the Knowledge Carried out of a Digital Learning Game?

Baker, Ryan

Scruggs, Richard

Pavlik, Philip I.

McLaren, Bruce M.

Liu, Ziyang

Abstract

Despite considerable advances in knowledge tracing algorithms, educational technologies that use this technology typically continue to use older algorithms, such as Bayesian Knowledge Tracing. One key reason for this is that contemporary knowledge tracing algorithms primarily infer next-problem correctness in the learning system, but do not attempt to infer the knowledge the student can carry out of the system, information more useful for teachers. The ability of knowledge tracing algorithms to predict problem correctness using data from intelligent tutoring systems has been extensively researched, but data from outcomes other than next-problem correctness have received less attention. In addition, there has been limited use of knowledge tracing algorithms in games, because algorithms that do attempt to infer knowledge from answer correctness are often too simple to capture the more complex evidence of learning within games. In this study, data from a digital learning game, *Decimal Point*, was used to compare ten knowledge tracing algorithms' ability to predict students' knowledge carried outside the learning system – measured here by posttest scores – given their game activity. All Opportunities Averaged (AOA), a method proposed by Scruggs, Baker, & McLaren (2020) was used to convert correctness predictions to knowledge estimates, which were also compared to the built-in

estimates from algorithms that produced them. Although statistical testing was not feasible for these data, three algorithms tended to perform better than the others: Dynamic Key-Value Memory Networks, Logistic Knowledge Tracing, and a multivariate version of Elo. Algorithms' built-in estimates of student ability underperformed estimates produced by AOA, suggesting that some algorithms may be better at estimating performance than ability. Theoretical and methodological challenges related to comparing knowledge estimates with hypothesis testing are also discussed.

Introduction

Knowledge tracing is a common way to use data from intelligent tutoring systems to gain insights about students' learning. One of the most important applications of knowledge tracing models is to inform teachers about students' knowledge (Ritter et al., 2016). Knowledge tracing has received a lot of study in recent years as newer algorithms have been developed which have been shown to be very accurate at predicting student performance. However, current learning software used at scale largely uses older knowledge tracing algorithms, rather than the most recent work in this area.

Recent work in knowledge tracing mostly focuses on inferring future problem correctness within the learning system being studied, treating better performance at this task as evidence of a better model (e.g., Choffin et al., 2019; Gervet et al., 2020; cf. Pelánek, 2017). However, as noted by Scruggs, Baker, & McLaren (2020), it is also important – perhaps more important -- to capture latent knowledge that transfers outside a learning system. Ultimately, teachers and other stakeholders need to know what knowledge a student has (Curry et al., 2016), not just how well they will perform on the next problem within the system. Early work on the Bayesian Knowledge Tracing (BKT) algorithm examined the model's ability to estimate how well students would perform outside a learning system (i.e., Corbett & Anderson, 1995; Corbett & Bhatnagar, 1997; Baker et al., 2010; Pardos et al., 2011), and some algorithms based on item response theory also provide estimates of latent knowledge (Klinkenberg, Straatemeier, & van der Maas, 2011; Wilson et al., 2016). By contrast, work over the last decade has produced increasingly complex models that attempt to fit item-to-item or new bottom-up skill-to-item mappings (i.e., Deep Knowledge Tracing (DKT) and Dynamic Key-Value Memory Networks (DKVMN); Piech et al., 2015; Zhang et al., 2017). While these models typically perform better at predicting

knowledge within the learning system (Zhang et al., 2017; Gervet et al., 2020), they no longer explicitly attempt to connect to external knowledge. To address this limitation, Scruggs and colleagues (2020) introduced a method (called "AOA") that extends these approaches to map them back to interpretable latent skills and tests the approach on post-tests external to the learning system. This method – averaging the predictions of student performance at each step of the students' work – was simple but functioned unexpectedly well at predicting performance on an external post-test.

In this paper, we extend this work by investigating whether the AOA method can also work for types of learning technologies where knowledge tracing has typically not been used, such as learning games. These environments are often too complex for classic algorithms such as BKT: in many games, a student's behavior can provide evidence about many competencies simultaneously, and there is not a 1:1 mapping between an action and a single skill, leading some researchers to propose machine learning (Rowe et al., 2022) or deep learning knowledge modeling approaches (e.g. Hooshyar et al., 2022). However, these contemporary algorithms cannot currently provide the information most useful to teachers. If AOA can be applied in learning games, then it may be possible to derive information about student knowledge from gameplay, in an actionable fashion that teachers can use. However, most current research on knowledge tracing uses large benchmark datasets from intelligent tutors (e.g., Choffin et al., 2019; Gervet et al., 2020; Zhang et al., 2017), with other learning environments receiving less attention (but there are exceptions, as discussed in the review of Abyaa et al., 2019; some notable cases include Lee et al., 2015; Pardos et al., 2013). It is not yet clear how well new knowledge tracing algorithms predict problem correctness – or other outcomes – using data from digital learning games or other types of learning environments.

Digital learning games are a popular way to teach many concepts, particularly in math and science (Mayer, 2019). As students interact with digital learning games, their actions can be recorded in log files. These log files can reflect very detailed learning processes which can further our understanding of how students learn (see e.g., Koedinger et al., 2010; 2013; Shute et al., 2009).

Clarke-Midura and Yudelson (2013) discuss the possibility of applying machine learning methods to infer students' understanding in digital learning games from such log data. Most subsequent research in this context focuses on direct predictions of posttest performance, without attempting to estimate student knowledge (e.g., Georgiadis et al., 2019; Ke, Parajuli, & Smith, 2019; Min et al., 2015), or assesses player understanding at individual time points (such as game rounds or levels) separately (e.g., Asbell-Clarke, Rowe, & Sylvan, 2013; Rowe et al., 2020). These studies (and others) demonstrate the feasibility of measuring knowledge from player behavior in a variety of types of games and learning genres. However, very few studies have modeled student knowledge in games using knowledge tracing methods which explicitly model how students' understanding grows and aggregate evidence of student understanding over time into a single overall measure of student knowledge on a skill or knowledge component. We are aware of only the work of Lee et al. (2015), which uses Bayesian Knowledge Tracing (BKT) to study students' learning in a physics game. They separately fit BKT on segments of gameplay, then build clusters using BKT's parameter estimates to better understand how students' knowledge emerges. As game-based learning and stealth assessment in games have become more common, understanding students' moment-by-moment learning as they interact with games has become more important (Owen & Baker, 2019).

In this paper, then, our goal is to investigate whether the AOA extension makes it possible for games to effectively infer students' knowledge that carries out of those games. In doing so, we compare contemporary algorithms extended with AOA to earlier algorithms that produce their own knowledge estimates – Bayesian Knowledge Tracing (BKT; Corbett & Anderson, 1995), Correct First Attempt Rate (Yu et al., 2010), Elo (Klinkenberg, Straatemeier, & van der Maas, 2011), Item Response Theory (de Ayala, 2009), Hierarchical Item Response Theory (Wilson et al., 2016), and Performance Factors Analysis (Pavlik, Cen, & Koedinger, 2009), to understand the relative contribution of the newer algorithms for this goal.

Extending Knowledge Tracing with All Opportunities Averaged (AOA)

While algorithms such as BKT and PFA produce their own estimates of students' skill level, many of the newer algorithms used in this paper only produce estimates of problem correctness. Problem correctness estimates were converted to skill estimates as in (Scruggs et al., 2020). For each algorithm, performance predictions were collected for each problem that each student attempted. Those predictions were then grouped by skill and averaged, giving a mean performance prediction for each skill, for each student. This method results in skill estimates produced by averaging correctness predictions on all of a student's opportunities to practice the skill, hence the name All Opportunities Averaged. This method was used to produce skill estimates for all algorithms (except CFAR), even including those algorithms that produce their own skill estimates, based on findings in (Scruggs et al., 2020), where this method outperformed several algorithms' own skill estimates. All estimates produced by averaging opportunities have a "-AOA" suffix in the results section.

Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT; Corbett & Anderson, 1995) is a state-based knowledge tracing algorithm where student learning is modeled as students transitioning from an unlearned state (not knowing the skill) to a learned state (knowing the skill) as they work on exercises. BKT models all skills separately and treats all items as being equally difficult. BKT also differs from all other algorithms in our study by being explicitly designed to infer latent knowledge (and predict test performance), although it also predicts next-item correctness. A BKT model is fit by empirically determining four probabilities: the chance of a student correctly answering an item from a skill they do not know (guess), incorrectly answering an item from a skill they do know (slip), initially knowing a skill, and transitioning between the unlearned and learned states.

In this study, BKT was implemented using code from Baker et al. (2010) to fit the parameters. As described in Baker, Corbett, and Aleven (2008), the parameters were bounded to avoid model degeneracy. All parameters had a floor of 0.01, guess and slip had a ceiling of 0.3, and the other two parameters had a ceiling of 0.99. Once fit, parameters were applied in Excel and final knowledge estimates and correctness predictions were collected. Although BKT does generate its own knowledge predictions, we apply AOA to convert its correctness predictions as well.

Deep Knowledge Tracing

Deep knowledge tracing (DKT; Piech et al., 2015) uses long short-term memory networks, a complex variant of recurrent neural networks, to predict correctness of student responses based on past activity. DKT does not provide estimates of student knowledge or skill performance, only predictions of correctness for each problem.

DKT was implemented using Yeung and Yeung's (2018) extensions to the original DKT method, which uses regularization to reduce occasional fluctuations in correctness prediction values and eliminate instances where predicted correctness decreased after students' correct answers or increased after incorrect answers. We used AOA to create knowledge estimates from DKT's problem correctness predictions.[1]

Dynamic Key-Value Memory Networks for Knowledge Tracing

Dynamic Key-Value Memory Networks (DKVMN) is an algorithm created by Zhang and colleagues (2017) which creates its own knowledge component (KC) mapping based on input data. It then estimates student mastery on these KCs and uses those estimates to predict student correctness. Unlike most algorithms that produce mastery estimates, DKVMN's estimates cannot be applied back to predefined skills in a straightforward manner due to its use of its own item-KC mapping. In this study, we implemented DKVMN using code from Zhang et al. (2017). After fitting DKVMN, we used AOA to translate its correctness predictions to knowledge estimates.

Performance Factors Analysis

Developed by Pavlik, Cen, and Koedinger (2009), Performance Factors Analysis (PFA) uses a logistic function to model and predict student performance based on students' successes and failures as they practice various skills. In this study, the algorithm was implemented in Python following the formulas in Pavlik et al. (2009), using SciPy's (Virtanen et al., 2020) BasinHopping optimizer to fit parameter estimates. Those parameters were applied in Excel; we then recorded individual correctness predictions and each student's final learning probability for each skill. Similarly to the other studied algorithms, in addition to using the final learning

[1] Using DKT, we were unable to calculate valid correctness predictions for 22 problem attempts, out of 68,033 attempts. Those invalid attempts were omitted.

probabilities as knowledge estimates, we used AOA to generate knowledge estimates from PFA's correctness predictions.

Item Response Theory

Item response theory (IRT), dating back to the 1950s, assumes that respondents' ability and item difficulty lie along a continuum. Individuals' correct and incorrect responses to items of varied difficulty are used to estimate their latent ability (de Ayala, 2009). As IRT focuses on analysis of test responses rather than learning activities, it generally assumes that students' ability does not change while responding to items. However, Wilson, Karklin, Han, and Ekanadham (2016) used a one-parameter IRT model and recomputed ability estimates after each attempt, with the resulting correctness predictions giving higher AUC than a DKT model on several large data sets. Wilson et al. (2016), noting that many items in online tutoring systems are built from a small number of templates, also found that a hierarchical IRT (HIRT) model with item templates that nested items within groups outperformed one-parameter IRT and DKT.

In this study, both IRT and HIRT were implemented using code from Wilson et al. (2016). Our dataset contained problem solving mini-games which were used as templates, giving a total of 24 different templates. Each template applied to two related skills (skills are discussed in the next section). As discussed in Wilson et al. (2016), HIRT's use of template labels means that it had access to more information than the other algorithms studied. Finally, IRT and HIRT produce estimates of students' ability on each skill, but we also applied AOA to generate knowledge predictions.

Elo

The Elo rating system, devised by Arpad Elo, is commonly used in chess and other games to estimate players' ability based on their wins and losses versus other players.

Klinkenberg, Straatemeier, and van der Maas (2011), noting mathematical similarities between Elo and the Rasch item response theory model, developed an algorithm that treats both students and items as players. Students that respond to items correctly are interpreted as having "won their match" against the item, thus enabling estimation of item difficulty and student ability. Since this algorithm can swiftly update estimates upon receiving new data, it is used in several adaptive learning systems (Pelánek, 2016).

In this study, Elo was implemented using code from Abdi et al. (2019). Two different variants of Elo were tested: single-concept and multivariate. In single-concept Elo, models were fit on each skill independently, while multivariate Elo included a global proficiency parameter for each student. Elo produces its own estimates of student ability, but as with the other algorithms, we also used AOA to convert its performance predictions to knowledge estimates.

Correct First Attempt Rate

Correct First Attempt Rate (CFAR) is a simple algorithm used by Yu et al. (2010) in the 2010 KDD Cup. It estimates student knowledge by computing students' average correctness over all problems that they have attempted so far in the given skill, only considering their first attempts at each problem. As CFAR is already similar to AOA, we do not average its estimates, only taking final values for each student, for each skill.

Logistic Knowledge Tracing

Logistic Knowledge Tracing (LKT; Pavlik, Eglington, & Harrell-Williams, 2021) is a flexible knowledge tracing framework that uses a variety of features in logistic regression-based models to predict student performance. The precise features used are chosen as part of the model design process, but they are typically linear or nonlinear functions that vary based on students'

prior outcomes or their time spent practicing. Pavlik et al. (2021) describes 25 different features which are currently available in the LKT package (Pavlik & Eglington, 2021).

In this study, a relatively simple LKT model was constructed by one of the authors. The model included intercepts for students and items and two features. The first feature, capturing learning for the KC, was a logarithmic version of the additive factors model feature, using the natural log of the count of prior observation plus one (Chi et al., 2011). The second feature, which adapts to performance on the KC, was based on an exponential recency weighted function of prior probability correct introduced as a feature in the recent-PFA (rPFA) work of Galyardt and Goldin (2015). Our version of this feature used an exponential recency weighted function of the prior logit. In this function successes and failures decay as in rPFA model, however, we compute log(decayed (success+1)/decayed (failiures+1)) whereas rPFA uses decayed successes/decayed (successes +3). This feature was chosen over the rPFA version since it is exactly centered on the logit value that would produce the prior performance observed. Other than the regression coefficients, the LKT model requires 1 additional non-linear parameter to characterize the recency function, which was optimized using LKT's built in L-BFGS-B search for non-linear feature parameters. The model did not produce skill estimates because each prediction depends on the item intercept, therefore AOA was used to translate its correctness predictions.

Participants, Data Collection and Algorithm Application

Data for this study was originally collected for a series of studies on teaching decimal concepts with *Decimal Point*, a digital learning game (Forlizzi et al., 2014; McLaren et al., 2017). The game is based on an amusement park metaphor, with the player moving between twenty-four different mini-games that teach different decimal concepts. The game has a narrative

in which alien characters have come to Earth and must learn about decimals; the student's job is to "teach" the characters while solving problems within each mini-game. *Decimal Point* is a game comprised of a variety of game features, including fantasy, non-competitive environment, and slow pace (Costello & Edmonds, 2007).

Students from middle schools in the United States were included in the original studies (McLaren et al., 2017), which took place across four semesters. Students completed a pretest (not used in the current study), played the learning game during their regular math classes for seven days, then completed a posttest and a delayed posttest (also not used). The original studies compared the game with a non-game control condition, but this study only includes the 500 students who were assigned to the game condition. During one semester, the original study examined the effect of erroneous examples on decimal understanding. 169 students received materials with an additional self-explanation subproblem on at least some of the problems; 61 of those students received an additional self-explanation subproblem on all problems.

The *Decimal Point* materials consisted of 48 game-based problems, each of which comprised several subproblems (referred to here as items), for a total of 297 items. For both the standard problem-solving and erroneous example conditions, after answering each problem, students were asked, via a multiple-choice question, to give advice to a hypothetical student solving that problem. The advice was essentially a "self-explanation" of how the problem was solved. For the erroneous example condition, before solving each problem, students were presented with a hypothetical student's incorrect attempt, then asked what reasoning error led to their mistake.

All materials and posttests were delivered through the MathTutor learning management system (Aleven, McLaren, & Sewall, 2009), which recorded all interactions. More information about the game materials is available in McLaren et al. (2017).

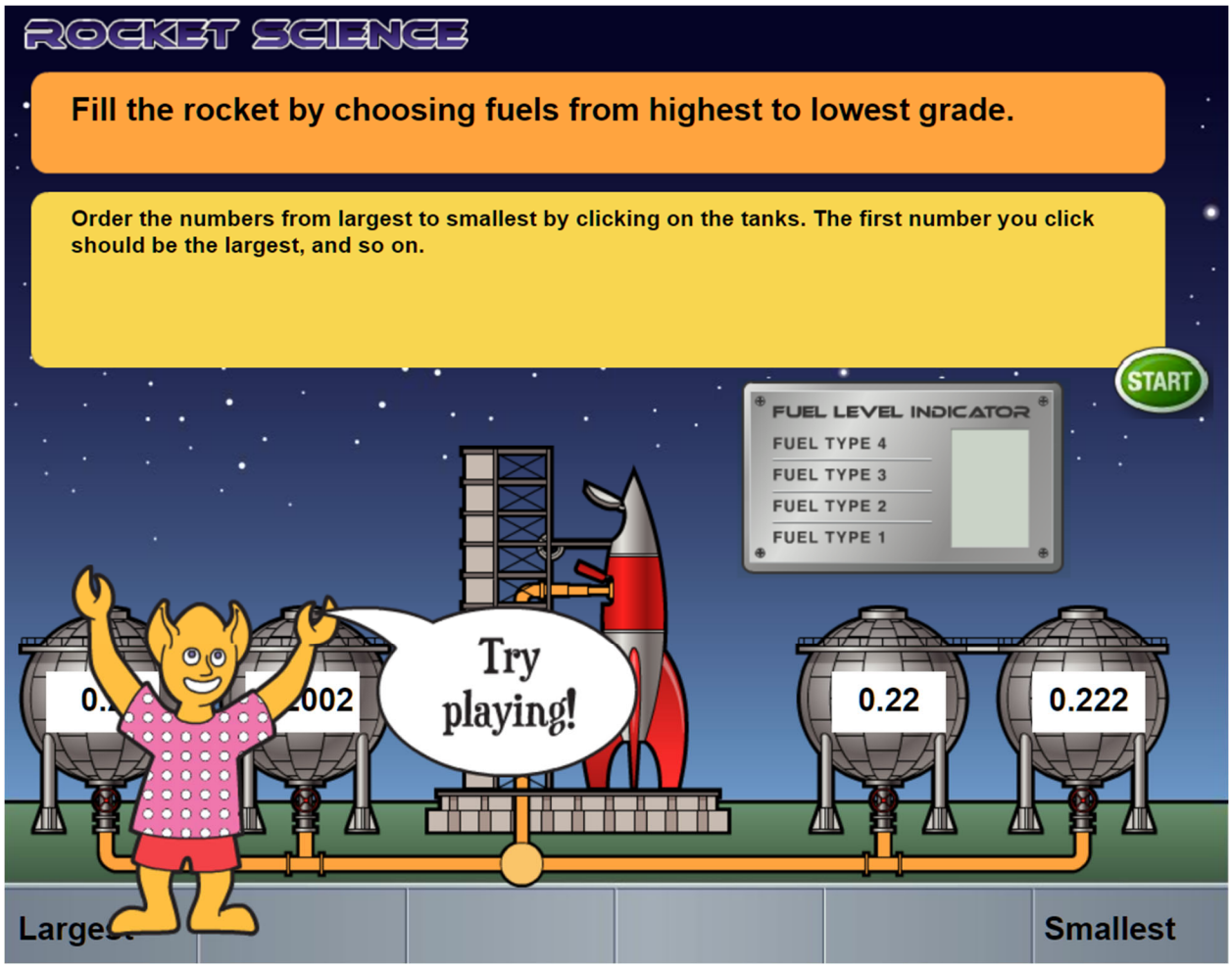

Figure 1. Decimal Point Rocket Science mini-game, asking students to demonstrate the Sorting skill.

The problems covered five different skills:

- Addition – add two decimal numbers together, entering the result and any applicable carried values.
- Bucket – compare several decimal numbers to a single criterion number, grouping the decimals that are larger or smaller than the criterion into two different buckets.
- Number Line – place a given decimal number on a number line.

- Sequence – continue a sequence of three given decimal numbers by entering the next two numbers in the sequence.
- Sorting – select from a group of decimal numbers in either ascending or descending order.

Some skills, such as Addition, ask students to enter multiple values before submitting their final answer; a mistake in any entered value resulted in the subproblem being incorrect. In this study, we treat each answer submission as a single attempt regardless of the number of values entered.

In addition, the mini-game format allows students to restart problems at will. In this study, we only used data from students' first attempts at subproblems, following standard practice in knowledge tracing research. For example, if a student began a problem with four subproblems, attempted two subproblems, restarted the problem, then attempted all four, we used the first two subproblem attempts from their first try, then the second two subproblem attempts from their second try.

As the advice and self-explanation questions focus more on conceptual understanding of the material than the rest of the subproblems, we treated them as distinct skills. This gave, for example, Addition, covering students' attempts to add decimal numbers, and Addition-Q, covering conceptual questions before or after addition subproblems. For more information on skills in this game, see Nguyen, Wang, Stamper, & McLaren (2019).

In total, our data set contained 68,033 student attempts at subproblems: 7,724 for Number Line, 7,066 for Number Line-Q, 3,767 for Bucket, 4,602 for Bucket-Q, 1,884 for Addition, 2,304 for Addition-Q, 7,640 for Sorting, 9,331 for Sorting-Q, 20,225 for Sequence, and 3,490 for Sequence-Q. The varying number of attempts relates to how the problems were presented in the

system. For example, Number Line attempts required students to place only one value on a number line before receiving feedback. Meanwhile, one Addition attempt required students to fill in a separate answer blank for each value in the presented problem.

After students completed the problems, their understanding was checked with a 37-item posttest, testing the five primary skills. The posttest contained 11 questions on Sorting, 8 on Addition, 7 on Sequence, 3 on Bucket, and 2 on Number Line, with the remaining 6 questions covering material that did not map to these skills. While the posttest did not contain any questions that were explicitly conceptual, its questions were more conceptual than the exercises and both the directly skill-based and conceptual activities in the game corresponded to the posttest. Thus, when making comparisons we averaged the primary skill predictions with the accompanying conceptual skill predictions (i.e., we averaged Sequence with Sequence-Q, and so on).

The process of fitting each algorithm, taking its knowledge estimates, and applying AOA to translate its correctness predictions to knowledge estimates was broadly similar for all algorithms. We used the entire dataset to fit each algorithm; if an algorithm expected multiple datasets (e.g., a test set and a training set) the same set was used for each. We did not hold out a test set from the game data, because this paper aims to evaluate performance on the (completely unseen) post-test given outside the learning system.

After each algorithm was fit, we collected its correctness predictions, which we processed with AOA to produce knowledge estimates. For BKT, CFAR, Elo, IRT, HIRT, and PFA, we also collected each skill's final knowledge estimate for each student. IRT, HIRT, and Elo all output z-scored final knowledge estimates, which we converted with a logistic transformation to values between 0 and 1 to be able to compare RMSE between algorithms.

Results

After producing estimates of latent knowledge from each algorithm for each student and each skill, we computed squared differences between each algorithm's estimate of each skill and students' actual score for that skill on the posttest. The Spearman correlations between each algorithm's within-tutor knowledge estimates and students' posttest performance on each skill are shown in Table 1. Table 2 depicts the root mean square error, and Figures 2 through 4 are density plots which show selected algorithms' accuracy in more detail.

	Addition	Bucket	NumLine	Sequence	Sorting
BKT	0.17	0.30	0.56	0.26	0.55
BKT-AOA	0.38	0.38	0.55	0.35	0.55
CFAR	0.28	0.40	0.57	0.39	0.56
DKT-AOA	0.31	0.39	0.55	0.32	0.55
DKVMN-AOA	0.51	0.42	0.59	0.45	0.61
Elo-M	0.38	0.39	0.55	0.39	0.56
Elo-M-AOA	0.50	0.44	0.59	0.46	0.60
Elo-S	0.25	0.37	0.54	0.36	0.54
Elo-S-AOA	0.31	0.39	0.57	0.39	0.56
HIRT	0.26	0.38	0.56	0.39	0.55
HIRT-AOA	0.28	0.36	0.53	0.39	0.53
IRT	0.26	0.38	0.56	0.39	0.55
IRT-AOA	0.27	0.35	0.53	0.39	0.53
LKT-AOA	**0.53**	0.43	**0.61**	**0.49**	**0.63**
PFA	0.50	**0.45**	0.56	0.39	0.58
PFA-AOA	0.36	0.40	0.52	0.36	0.52
Average	0.35	0.39	0.56	0.39	0.56

Table 1. Spearman correlations between predictions and posttest scores. Higher correlations for each skill are more heavily shaded, with the highest for each skill in bold.

	Addition	Bucket	NumLine	Sequence	Sorting
BKT	0.34	0.39	0.37	0.41	**0.29**
BKT-AOA	0.29	0.40	0.37	0.31	0.31
CFAR	0.33	0.40	**0.35**	0.31	0.30
DKT-AOA	0.33	**0.38**	0.37	0.31	**0.29**
DKVMN-AOA	0.27	0.39	0.36	0.30	**0.29**
Elo-M	0.32	0.45	0.39	0.25	0.34
Elo-M-AOA	**0.26**	0.39	**0.35**	0.30	**0.29**
Elo-S	0.35	0.45	0.39	0.27	0.34
Elo-S-AOA	0.30	0.40	**0.35**	0.31	0.30
HIRT	0.32	0.44	0.39	**0.24**	0.34
HIRT-AOA	0.29	0.40	0.37	0.31	0.31

IRT	0.32	0.44	0.39	0.25	0.34
IRT-AOA	0.30	0.40	0.37	0.30	0.31
LKT-AOA	**0.26**	0.39	**0.35**	0.30	**0.29**
PFA	0.45	0.43	0.45	0.56	0.35
PFA-AOA	0.43	**0.38**	0.45	0.55	0.35
Average	0.32	0.41	0.38	0.33	0.31

Table 2. RMSE values between predictions and posttest scores. Lower values for each skill are more heavily shaded, with the lowest for each skill in bold.

Looking at tables 1 and 2, it is clear that while some algorithms produce estimates that are generally closer to the posttest, no algorithm definitively outperforms the others on all skills, across both metrics. There were also dramatic differences in performance on different skills. Looking at Spearman correlations, all algorithms performed similarly on Number Line and, to a lesser extent, on Sorting, both led by LKT-AOA. Addition, with the fewest items, saw many of the single-skill algorithms struggle, while algorithms that were able to use data from other skills tended to perform better. PFA had the highest Spearman correlation on the Bucket skill, although Elo-M-AOA, LKT-AOA, and DKVMN-AOA all had high correlations as well. LKT-AOA, Elo-M-AOA, and DKVMN-AOA also performed much better than all other algorithms on Sequence. LKT-AOA, multivariate Elo AOA, and DKVMN-AOA correlated best with the posttest after averaging across all skills. Although AOA did not help in all cases, it appeared to help in more cases than it hurt.

RMSE showed slightly more mixed results, with nine different algorithms leading on at least one skill (due to ties between algorithms). PFA struggled across many skills according to RMSE; nearly all of its final knowledge estimates were above 0.99 or below 0.01, leading to large RMSE values when predicting the posttest. AOA still appeared to lead to better performance for most algorithms, leading to lower RMSEs in 70% of cases, but on Sequence, the best-performing algorithms were all using their original skill estimates instead of AOA. After

averaging across skills, LKT-AOA, Elo-M-AOA, and DKVMN-AOA had the lowest RMSE values, just as for Spearman.

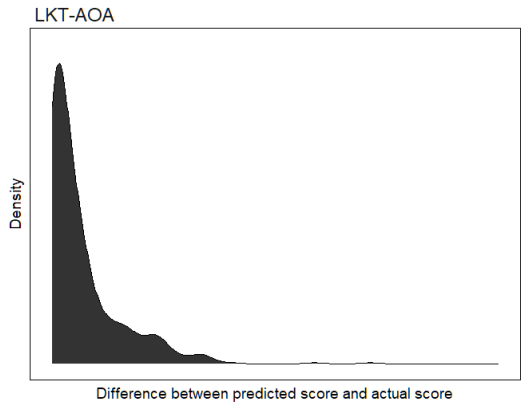

Figure 2. Density plot of LKT-AOA on Addition.

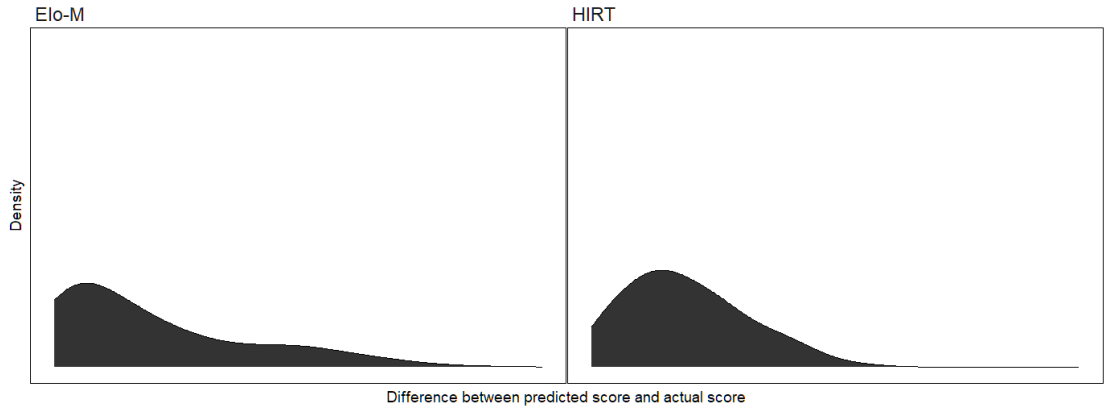

Figure 3. Density plot of multivariate Elo and HIRT on Bucket.

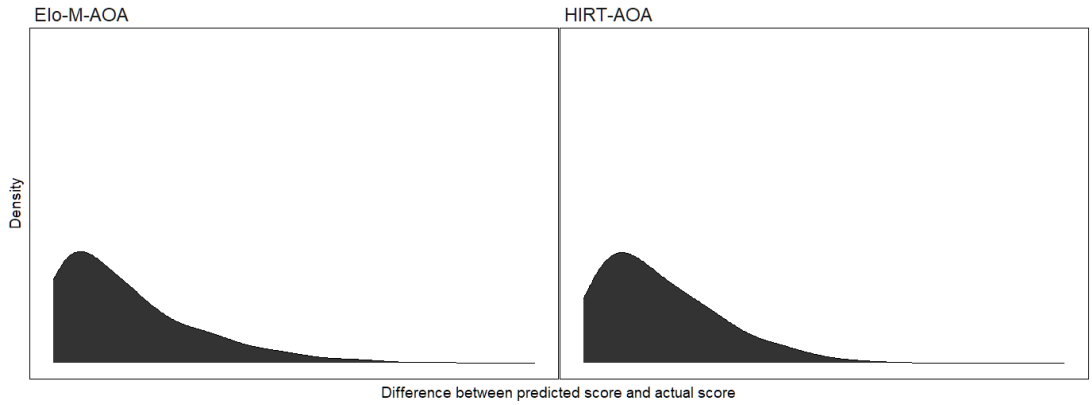

Figure 4. Density plot of multivariate Elo-AOA and HIRT-AOA on Bucket.

To offer more detail than simply RMSE values, the density plots in figures 2 through 4 offer graphical representations of how closely a selection of algorithms were able to predict the posttest for each student. Higher peaks at the left of a plot indicates that for that skill, more of that algorithm's predictions were close to students' actual posttest scores. Rightward tails show the predictions which differed greatly from posttest scores.

Figure 2 depicts the strong performance of LKT-AOA in estimating students' skill levels for the Addition skill. The peak is high and very close to the Y-axis, with a relatively short tail, meaning that all estimates were quite close to students' post-test scores. By contrast, Figure 3 shows multivariate Elo and HIRT's poorer estimates on the Bucket skill. The peaks are lower and farther from the Y-axis, while the tails are longer and heavier, meaning that more estimates were wider of the mark. Note that although these two algorithms' RMSE differed by only 0.01 for this skill, their density plots look quite different. HIRT's peak is farther from the Y-axis, but it has a shorter tail: its estimates frequently differ moderately from posttest scores, but they do so more consistently and are rarely extremely wrong. These two cases have very different use implications when making inferences about students in real-time – a model that is usually moderately wrong may be less likely to make badly inappropriate decisions than a model that is either spot on or completely inaccurate. Finally, Figure 4 shows estimates from HIRT and multivariate Elo after applying AOA, again on the Bucket skill. Averaging opportunities led to much more accurate estimates, with fewer estimates that were way off – the peaks of the graph are higher, with lighter tails. The peaks are also closer to the Y-axis, particularly for HIRT.

Statistical Analysis: Difficult to Conduct

There is a natural desire to conduct statistical comparisons to decide whether one algorithm performs statistically significantly better than another algorithm, across skills.

Unfortunately, the nature of these data makes conventional hypothesis testing very challenging. When using squared differences between skill estimates and posttest scores, the resulting values violate nearly every assumption of classical MANOVA: they exhibit large amounts of positive skew, contain both univariate and multivariate outliers, they are both univariate and multivariate non-normal, nonlinear relationships between skills are common, and they have they heterogenous between-group variance and covariance. While MANOVA is generally robust to some assumption violations, particularly for large sample sizes and balanced designs (Vallejo & Ato, 2012), researchers rarely test violations of more than two assumptions simultaneously and never test with as many groups as are present here.

There exist many robust or nonparametric variants of classical MANOVA as well (see citations in Finch & French, 2013; Konietschke, Bathke, Harrar & Pauly, 2015), but these approaches still have assumptions, related to homogenous variance (e.g., Anderson, 2001) or error distributions (e.g., Bathke et al., 2018), which are also violated by these data. Indeed, once data differ in so many ways, it may not be appropriate to statistically compare them based on a single value (group means or medians). For example, one algorithm may have larger overall root mean square error than another algorithm, but may be preferable due to having fewer extremely inaccurate predictions, e.g., causing a student to be advanced for having achieved mastery, when they are on track to do very poorly on the post-test. Finally, while we could compare skills individually, the number of statistical tests performed would lead to non-significant results after applying a post-hoc control. For these reasons, we only present descriptive statistics.

Finding appropriate statistical ways to analyze and compare skill estimates from different algorithms will be a significant challenge for future work in this field. While the overwhelming majority of work on knowledge tracing focuses on within-system performance prediction,

measuring and supporting the development of more latent knowledge and skill – knowledge and skill that carries out of the learning system – is the goal of most adaptive learning systems. As such, it's important to be able to compare skill estimates and performance at measuring skill transferred outside the system, to determine which algorithms are more accurate. Doing so in future work requires addressing several challenges, both statistical and theoretical.

From a statistical perspective, there are a variety of metrics that might be used to evaluate model fit (see, e.g., Pelánek, 2015; Schunn & Wallach, 2005). Effenberger and Pelánek (2020) discuss how methodological choices – e.g., which performance metrics are used and exactly how predictions are combined from cross-validation folds – impact results in knowledge tracing tasks.

In addition, because nearly all learning systems contain multiple skills, researchers must decide how to combine estimates from different skills (see also Effenberger & Pelánek, 2020 for a related discussion of combining model fit on easy, intermediate, and difficult exercises and Pelánek, 2020, for a broader discussion of skills and domains). The goal of combining multiple estimates leads naturally to an approach like MANOVA, but the operation of MANOVA combines dependent variables so as to maximize between-group differences. In effect, this creates a weighting of skill importance which – as it will differ from study to study – prevents the comparison of results from one study to another, even if the same data sets are used.

Assuming that all skills in a combined comparison should have equal weight is not a foolproof solution either. In this study, the algorithms used performed very differently on the different skills, with seven of the sixteen variants leading for at least one skill, on at least one metric. However, the skills had very different amounts of data available, with Sequence having more than ten times the attempts of Addition. Whether these skills should be seen as equally important is an open question, and likely depends on the goal of developing the algorithm.

Finally, as nearly all learning systems cover multiple skills, it is likely not appropriate for an algorithm's strong performance on some skills to offset its poor performance on others – users will expect the same level of accuracy throughout the system. When researchers compare knowledge estimates across multiple skills, then, it is important to show a full picture of the comparison being conducted. In addition to whatever final statistic is presented, means, minimums, and standard deviations should be reported across skills for whichever metrics are chosen. This will illustrate algorithms' consistency and will help make algorithms' outputs more understandable (see related discussion in Webb et al., 2021).

Discussion and Conclusion

This study demonstrates that a range of modern knowledge tracing algorithms can produce good out-of-system knowledge predictions on predefined skills, in a game context. While the algorithms used in this study rarely achieved posttest correlations as high as in past work studying these algorithms within an intelligent tutor (Scruggs et al., 2020), the correlations here were generated with fewer problem attempts from each skill. This study did not show the same dramatic benefits for AOA as that prior work, but AOA still enabled LKT and DKVMN to generate excellent estimates for our predefined skills. AOA also appeared to improve performance both single- and multiple-concept Elo and BKT, if slightly, but had little impact on the quality of the predictions made by IRT or HIRT with this data set and actually led to PFA having worse Spearman correlations with the posttest.

Although the results varied notably among the five skills, three of the sixteen algorithms or variants tested, LKT-AOA, DKVMN-AOA, and multivariate Elo-AOA, outperformed the average performance of all algorithms on every skill on both Spearman correlation coefficients and RMSE. LKT-AOA performed particularly well, leading the field five times.

Somewhat surprisingly, algorithms that generated their own skill estimates tended to underperform. BKT never outperformed the average Spearman correlation and only beat the average RMSE twice. IRT, HIRT, standard Elo, and multivariate Elo combined only outperformed the average in two cases for Spearman and in four cases for RMSE, all on Sequence. However, applying AOA to multivariate Elo's correctness predictions yielded knowledge estimates that outperformed the average across both metrics on every skill. This suggests that algorithms' knowledge estimates and performance predictions should be evaluated separately – an algorithm that produces good performance predictions may produce poor knowledge estimates and vice versa. If this finding is replicated, it could have significant impacts on algorithm choices in adaptive learning systems where it is important to both predict performance and estimate knowledge.

Our study also leads to conclusions about the individual algorithms studied. BKT performed poorly on Sequence, the skill with the most items. This somewhat echoes its lackluster showing in past papers involving both post-test prediction (i.e., Pardos et al., 2011; Scruggs et al., 2020) and within-system performance prediction (Gervet et al., 2020). Scruggs and colleagues (2020) theorized that BKT performed poorly in that paper due to overpracticing leading to inflated final knowledge estimates, but only 5% of Sequence estimates in this study were at or above 0.95. In fact, BKT produced more final estimates above 0.95 on the other skills, where it did better. BKT's interpretable structure makes it useful for a range of discovery with models analyses (Beck et al., 2008; Baker et al., 2010, 2018), but it generally underperforms at actual performance prediction, whether in-system or externally.

Like BKT, PFA may have done poorly because it has a similarly independent variable structure, with estimates of learning focused on KCs and ignoring item level differences. The

success of the model created in LKT is likely in part due to using an intercept to characterize item difficulty. This intercept controls for the difficulty of items during the process of model fitting, allowing better capture of skill-based variability. This results in a better-fitting model due to the fine control provided by the item difficulties, which capture substantial variance. Using the log of the opportunities was also likely important to capture decelerating learning.

IRT and HIRT never achieved both a good Spearman correlation and a good RMSE on any skill, although they did achieve acceptable values on one of these metrics for some skills. HIRT never performed notably better than IRT, suggesting that our mini-games may not be as different as the templates used in Wilson et al. (2016). Other than multivariate Elo, which used data from other skills, the IRT variants also struggled on Addition; they may need more data to produce good predictions.

Unlike in (Scruggs et al., 2020), the two deep learning algorithms used were never top performers with this game-based data set. While DKVMN performed quite well overall, it only tied for lowest RMSE once and never achieved a top Spearman value. DKT did worse; although it tied for the lowest RMSE twice, its Spearman values were lackluster. This relative underperformance may be explained by the findings of Gervet et al. (2020), who find that DKT performs better on larger datasets. Although our data set was not appreciably smaller than Scruggs et al. (2020), with approximately 68,000 attempts vs 70,500, this data set contained one additional skill and thus had fewer attempts per skill, with many attempts belonging to the Sequence skill.

Recent work in knowledge tracing has focused on algorithms' ability to predict performance in large datasets from intelligent tutors. However, many students currently learn in game-based systems that are more engaging than traditional tutors. Showing that knowledge

tracing algorithms generate accurate estimates of students' knowledge from game data would support the use of such games as educational tools and make it easier to design stealth assessments in games. This study provides evidence for one game, but more research is needed to validate these results across different types and quality of games.

Although modern knowledge tracing algorithms can fit datasets very well, accurately predicting problem correctness, few of these algorithms have achieved widespread use in learning systems, in part because they do not provide useful information for teachers. The AOA extension makes it possible to map these systems' estimates of next-problem correctness back to more interpretable and usable latent skill estimates. In this paper we investigate the quality of these estimates, grounding our analysis in a post-test of interpretable skills given outside the learning system and describe some of the methodological and theoretical challenges to conducting comparisons of knowledge estimates. We hope that future researchers will continue looking beyond problem correctness and will consider these issues when comparing knowledge estimates. As adaptive learning systems begin adopting more advanced contemporary algorithms, we need to determine which applications each algorithm is best for, and what their full profile of strengths and weaknesses are.

Data Availability

The data that support the findings of this study are openly available in the CMU DataShop at https://pslcdatashop.web.cmu.edu/Project?id=67.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by grant NSF#DRL-1661121.

References

Abdi, S., Khosravi, H., Sadiq, S., & Gasevic, D. (2019). A multivariate Elo-based learner model for adaptive educational systems. *Proceedings of the 12th International Conference on Educational Data Mining*, 228–233. Montreal, Canada.

Abyaa, A., Khalidi Idrissi, M., & Bennani, S. (2019). Learner modelling: Systematic review of the literature from the last 5 years. *Educational Technology Research and Development*, *67*(5), 1105–1143. https://doi.org/10.1007/s11423-018-09644-1

Aleven, V., McLaren, B.M., & Sewall, J. (2009). Scaling up programming by demonstration for intelligent tutoring systems development: An open-access website for middle school mathematics learning. IEEE Transactions on Learning Technologies, 2(2), 64-78.

Asbell-Clarke, J., Rowe, E., & Sylvan, E. (2013). Assessment design for emergent game-based learning. *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, 679–684. Paris, France: Association for Computing Machinery. https://doi.org/10.1145/2468356.2468476

Baker, R.S.J.d., Corbett, A.T., Aleven, V. (2008). More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. Proceedings of the 9th International Conference on Intelligent Tutoring Systems, 406-415.

Baker, R.S.J.d., Corbett, A.T., Gowda, S.M., Wagner, A.Z., MacLaren, B.M., Kauffman, L.R., Mitchell, A.P., Giguere, S. (2010). Contextual slip and prediction of student performance after use of an intelligent tutor. Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization, 52-63.

Baker, R.S., Gowda, S.M., Salamin, E. (2018) Modeling the Learning That Takes Place Between Online Assessments. Proceedings of the 26th International Conference on Computers in Education, 21-28.

Beck, J. E., Chang, K. M., Mostow, J., & Corbett, A. (2008, June). Does help help? Introducing the Bayesian Evaluation and Assessment methodology. In *International conference on intelligent tutoring systems* (pp. 383-394). Springer, Berlin, Heidelberg.

Chi, M., Koedinger, K.R., Gordon, G., Jordan, P., & VanLehn, K. (2011). Instructional factors analysis: A cognitive model for multiple instructional interventions. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero , and J. Stamper (Eds.), *4th Int. Conf. Educational Data Mining* (pp. 61–70). Eindhoven, The Netherlands.

Choffin, B., Popineau, F., Bourda, Y., Vie, J.-J. (2019). DAS3H: Modeling student learning and forgetting for optimally scheduling distributed practice of skills. In *Proceedings of the 12th International Conference on Educational Data Mining*, C. F. Lynch, A. Merceron, M. Desmarais, and R. Nkambou, Eds. 29–39.

Clarke-Midura, J., & Yudelson, M. V. (2013). Towards Identifying Students' Causal Reasoning Using Machine Learning. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Artificial Intelligence in Education* (pp. 704–707). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-39112-5_93

Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253–278. https://doi.org/10.1007/BF01099821

Costello, B. & Edmonds, E. (2007) A study in play, pleasure and interaction design. *Proceedings of the 2007 conference on Designing pleasurable products and interfaces*, 22-25. ACM.

Curry, K. A., Mwavita, M., Holter, A., & Harris, E. (2016). Getting assessment right at the classroom level: Using formative assessment for decision making. *Educational Assessment, Evaluation and Accountability*, *28*(1), 89-104

De Ayala, R. J. (2009). *The theory and practice of item response theory*. New York: Guilford Press.

Effenberger, T., & Pelánek, R. (2020). Impact of Methodological Choices on the Evaluation of Student Models. In I. I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, & E. Millán (Eds.), *Artificial Intelligence in Education* (pp. 153–164). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-52237-7_13

Finch, H., & French, B. (2013). A Monte Carlo Comparison of Robust MANOVA Test Statistics. *Journal of Modern Applied Statistical Methods*, *12*(2), 35–81. https://doi.org/10.22237/jmasm/1383278580

Forlizzi, J., McLaren, B. M., Ganoe, C., McLaren, P. B., Kihumba, G., & Lister, K. (2014). Decimal point: Designing and developing a digital learning game to teach decimals to middle school students. 8th European Conference on Games-Based Learning: ECGBL2014, 128–135.

Galyardt, A. & Goldin, I. (2015). Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining, 7*(2), 83–108 https://doi.org/10.5281/zenodo.3554671.

Georgiadis, K., van Lankveld, G., Bahreini, K., & Westera, W. (2019). Learning Analytics Should Analyse the Learning: Proposing a Generic Stealth Assessment Tool. *2019 IEEE Conference on Games (CoG)*, 1–8. https://doi.org/10.1109/CIG.2019.8847960

Gervet, T., Koedinger, K., Schneider, J., & Mitchell, T. (2020). When is Deep Learning the Best Approach to Knowledge Tracing? *Journal of Educational Data Mining*, *12*(3), 31–54. https://doi.org/10.5281/zenodo.4143614

Hooshyar, D., Huang, Y. M., & Yang, Y. (2022). GameDKT: Deep knowledge tracing in educational games. *Expert Systems with Applications*, *196*, 116670.

Klinkenberg, S., Straatemeier, M., & van der Maas, H. L. J. (2011). Computer adaptive practice of Maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, *57*(2), 1813–1824. https://doi.org/10.1016/j.compedu.2011.02.003

Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining. Boca Raton, FL: CRC Press.

Koedinger, K.R., Stamper, J.C., Leber, B., & Skogsholm, A. (2013). LearnLab's DataShop: A data repository and analytics tool set for Cognitive Science. *Topics in Cognitive Science*. Doi: 10.1111/tops.12035

Konietschke, F., Bathke, A. C., Harrar, S. W., & Pauly, M. (2015). Parametric and nonparametric bootstrap methods for general MANOVA. *Journal of Multivariate Analysis*, *140*, 291–301. https://doi.org/10.1016/j.jmva.2015.05.001

Lee, H.-S., Gweon, G.-H., Dorsey, C., Tinker, R., Finzer, W., Damelin, D., … Lord, T. (2015). How does Bayesian knowledge tracing model emergence of knowledge about a mechanical system? *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge – LAK '15*, 171–175. Poughkeepsie, New York: ACM Press. https://doi.org/10.1145/2723576.2723587

Mayer, R.E. (2019) Computer Games in Education. *Annual Review of Psychology, 70*, 531-549, http://dx.doi.org/10.1146/annurev-psych-010418-102744

McLaren, B., Adams, D., Mayer, R., & Forlizzi, J. (2017). A Computer-Based Game that Promotes Mathematics Learning More than a Conventional Approach. International Journal of Game-Based Learning, 7, 36–56. https://doi.org/10.4018/IJGBL.2017010103

Min, W., Frankosky, M. H., Mott, B. W., Rowe, J. P., Wiebe, E., Boyer, K. E., & Lester, J. C. (2015). DeepStealth: Leveraging Deep Learning Models for Stealth Assessment in Game-Based Learning Environments. In C. Conati, N. Heffernan, A. Mitrovic, & M. F. Verdejo (Eds.), *Artificial Intelligence in Education* (pp. 277–286). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-19773-9_28

Pardos, Z. A., Gowda, S. M., Baker, R.S.J.d., Heffernan, N. T. (2011) Ensembling Predictions of Student Post-Test Scores for an Intelligent Tutoring System. Proceedings of the 4th International Conference on Educational Data Mining, 189-198.

Pavlik, P.I. & Eglington, L.G. (2021). LKT: Logistic Knowledge Tracing. R package version 1.0. https://CRAN.R-project.org/package=LKT

Pavlik, P.I., Eglington, L. G., & Harrell-Williams, L. M. (2021). Logistic Knowledge Tracing: A Constrained Framework for Learner Modeling. ArXiv:2005.00869 [Stat]. Retrieved from http://arxiv.org/abs/2005.00869

Pavlik, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis –A New Alternative to Knowledge Tracing. Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling, 531–538. Amsterdam, The Netherlands, The Netherlands: IOS Press. Retrieved from http://dl.acm.org/citation.cfm?id=1659450.1659529

Pelánek, R. (2015). Metrics for Evaluation of Student Models. *Journal of Educational Data Mining*, *7*(2), 1–19. https://doi.org/10.5281/zenodo.3554665

Pelánek, R. (2016). Applications of the Elo rating system in adaptive educational systems. *Computers & Education*, *98*, 169–179. https://doi.org/10.1016/j.compedu.2016.03.017

Pelánek, R. (2017). Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, *27***,** 313–350. https://doi.org/10.1007/s11257-017-9193-2

Pelánek, R. (2020). Managing items and knowledge components: Domain modeling in practice. *Educational Technology Research and Development*, *68*(1), 529–550. https://doi.org/10.1007/s11423-019-09716-w

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, 505–513. MIT Press.

Ritter, S., Yudelson, M., Fancsali, S.E., Berman, S.R. (2016). How Mastery Learning Works at Scale. *Proceedings of the 3rd ACM Conference on Learning @ Scale,* 71-79.

Rowe, E., Asbell-Clarke, J., Bardar, E., Almeda, Ma. V., Baker, R., Scruggs, R., & Gasca, S. (2020). Advancing Research in Game-Based Learning Assessment: Tools and Methods for Measuring Implicit Learning (E. Kennedy & Y. Qian, Eds.). https://doi.org/10.4018/978-1-7998-1173-2.ch006

Sao Pedro, M.A., Baker, R.S.J.d., Gobert, J., Montalvo, O. Nakama, A. (2013) Leveraging Machine-Learned Detectors of Systematic Inquiry Behavior to Estimate and Predict Transfer of Inquiry Skill. User Modeling and User-Adapted Interaction, 23 (1), 1-39.

Schunn, C. D., & Wallach, D. (2005). Evaluating goodness-of-fit in comparison of models to data. *Psychologie Der Kognition: Reden and Vorträge Anlässlich Der Emeritierung von Werner Tack*, 115–154.

SciPy 1.0 Contributors, Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., … van Mulbregt, P. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Scruggs, R., Baker, R.S., & McLaren, B.M. (2020). Extending deep knowledge tracing: Inferring interpretable knowledge and predicting post-system performance. In: So, H. J. et al. (Eds.) Proceedings of the 28th International Conference on Computers in Education (ICCE 2020)

Shute, V., Ventura, M., Bauer, M., & Zapata-Rivera, D. (2009). Melding the Power of Serious Games and Embedded Assessment to Monitor and Foster Learning: Flow and Grow. In U. Ritterfield, M. J. Cody, & P. Vorderer (Eds.), *Serious Games: Mechanisms and Effects* (Vol. 1, pp. 295–321).

Vallejo, G., & Ato, M. (2012). Robust tests for multivariate factorial designs under heteroscedasticity. *Behavior Research Methods*, *44*(2), 471–489. https://doi.org/10.3758/s13428-011-0152-2

Webb, M. E., Fluck, A., Magenheim, J., Malyn-Smith, J., Waters, J., Deschênes, M., & Zagami, J. (2021). Machine learning for human learners: Opportunities, issues, tensions and threats. *Educational Technology Research and Development*, *69*(4), 2109–2130. https://doi.org/10.1007/s11423-020-09858-2

Wilson, K. H., Karklin, Y., Han, B., & Ekanadham, C. (2016). Back to the Basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation. *ArXiv:1604.02336 [Cs]*. Retrieved from http://arxiv.org/abs/1604.02336

Yeung, C.-K., & Yeung, D.-Y. (2018). Addressing two problems in deep knowledge tracing via prediction-consistent regularization. *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, 1–10. London, United Kingdom: Association for Computing Machinery. https://doi.org/10.1145/3231644.3231647

Yu, H., Lo, H., Hsieh, H., Lou, J., Mckenzie, T. G., Chou, J., … Weng, J. (2011). Feature engineering and classifier ensemble for KDD Cup 2010. *In JMLR Workshop and Conference Proceedings*.

Zhang, J., Shi, X., King, I., & Yeung, D.-Y. (2017). Dynamic Key-Value Memory Networks for Knowledge Tracing. *Proceedings of the 26th International Conference on World Wide Web*, 765–774. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3038912.3052580