A Framework for Multifaceted Evaluation of Student Models

Yun Huang
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA, USA
yuh43@pitt.edu

José P. González-Brenes
Pearson Research &
Innovation Network
Philadelphia, PA, USA
jose.gonzalez-brenes@pearson.com

Rohit Kumar
Speech, Language and Multimedia
Raytheon BBN Technologies
Cambridge, MA, USA
rkumar@bbn.com

Peter Brusilovsky
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA, USA
peterb@pitt.edu

ABSTRACT

Latent variable models, such as the popular Knowledge Tracing method, are often used to enable adaptive tutoring systems to personalize education. However, finding optimal model parameters is usually a difficult non-convex optimization problem when considering latent variable models. Prior work has reported that latent variable models obtained from educational data vary in their predictive performance, plausibility, and consistency. Unfortunately, there are still no unified quantitative measurements of these properties. This paper suggests a general unified framework (that we call Polygon) for multifaceted evaluation of student models. The framework takes all three dimensions mentioned above into consideration and offers novel metrics for the quantitative comparison of different student models. These properties affect the effectiveness of the tutoring experience in a way that traditional predictive performance metrics fall short. The present work demonstrates our methodology of comparing Knowledge Tracing with a recent model called Feature-Aware Student Knowledge Tracing (FAST) on datasets from different tutoring systems. Our analysis suggests that FAST generally improves on Knowledge Tracing along all dimensions studied.

Keywords

Student Modeling, Knowledge Tracing, parameter estimation, Identifiability, Model Degeneracy

1. INTRODUCTION

Adaptive tutoring systems often rely on student models to trace the progress of student knowledge to personalize instruction. Such student models are usually latent variable models with the state of student knowledge as the latent variable. However, finding optimal model parameters is usually a difficult non-convex optimization problem for latent variable models. Moreover, in the context of tutoring systems, even global optimum model parameters may not be interpretable (or plausible). Knowledge Tracing [4] is one such latent variable model that has been widely used, and different properties of its estimated parameters have been presented in many previous studies: predictive performance [6], plausibility [1, 6, 19], and consistency [2, 6, 16, 19, 9]. Unfortunately, there are still no unified quantitative measurements of these properties. If prediction of student performance is our only goal, this need is less urgent, since we can simply pick a model according to classification metrics. However, parameters with varying properties might have different inferences about knowledge, which may result in different tutoring decisions that can have a large impact on students. To illustrate, we show examples where two models that both belong to Knowledge Tracing are fitted from the same data, and where predictive performance is not sufficient to pick a good model:

- One model with higher predictive performance asserts that student knowledge decreases with correct practices, while the other model asserts the opposite. In such cases, the former model will suggest continuing practicing even if students get a lot of correct answers in a row, while the latter will suggest moving to other skills in a shorter amount of time.

- Two models have the same predictive performance, yet one asserts that about 20 practices are required to reach mastery of a skill, while the other asserts that only about 3 practices are enough. In such cases, a student needs to practice a lot under the former model, but under the latter model, students can move to learning other skills more quickly.

In the first example, the more predictive model lacks plausibility; in the second example, two models lack consistency, even though they have the same predictive performance. As a result, we advocate that a student model should be examined from dimensions besides predictive performance. We propose a unified quantitative framework, called Polygon, for the multifaceted evaluation and comparison of student models. The framework suggests novel metrics to quantify the properties of a student model along multiple dimensions, including predictive performance, plausibility, and consistency. Polygon is designed for general latent variable models that model latent student knowledge and is domain-independent. In the present work, we demonstrate how we apply Polygon to evaluate and compare classic Knowledge Tracing with a recent generalized model called Feature-Aware Student Knowledge Tracing (FAST) [8] in four different domains. Section 2 reviews some latent variable student models and prior work examining their properties; Section 3 describes our Polygon framework and metrics; Section 4 studies the relationship among these metrics and compares Knowledge Tracing with FAST; Section 5 concludes the work.

2. BACKGROUND

2.1 Latent Variable Student Models

We now review two effective latent variable models for predicting student performance and inferring student knowledge: Knowledge Tracing [4] and Feature-Aware Student Knowledge Tracing (FAST) [8]. Knowledge Tracing uses Hidden Markov Models to model student knowledge as binary latent variables (either learned or unlearned), given the observed practice performance (correct or incorrect) and using four parameters: Init (initial knowledge level), Learn (learning rate), Guess, and Slip. We learn the parameters of Knowledge Tracing using the Expectation Maximization algorithm. A recent model FAST incorporates features into Knowledge Tracing by replacing the binomial distributions by logistic regression distributions. It encodes contextual information as features for the original Knowledge Tracing parameters. It allows flexible features to affect student performance or knowledge directly. For simplicity, we use features in all four parameters in the study. FAST trains feature coefficients jointly with other parameters using the Expectation Maximization with Features algorithm [3]. This algorithm keeps the original E-step and replaces the M-step by training a weighted regularized logistic regression using a gradient-based search algorithm (LBFGS). While FAST has been shown to outperform Knowledge Tracing in many prediction tasks, we are interested in comparing it with Knowledge Tracing in other dimensions.

2.2 Prior Work Examining Properties of Knowledge Tracing

Prior work has examined Knowledge Tracing models from predictive performance, plausibility, and consistency. We now review previous studies in each dimension.

Predictive Performance. Measurements of predictive performance have been broadly applied to evaluate student models. Prior studies have shown several problems with parameter estimation for Knowledge Tracing, which predictive performance metrics often fail to detect [2, 16, 7]. We examine this traditional dimension in more depth for both Knowledge Tracing and FAST, and complement it in other dimensions, including plausibility and consistency.

Plausibility. Interpretability of a model is a desire property because it allows for better scientific claims and practical applications. Prior studies have used external measurements for validating the plausibility of fitted parameters, such as pre-test scores [6], exercise scores [4], or some domain-specific measurements [2]. However, such external resources are not always available. Many studies also examined plausibility by internal validity. Learning curves plotted using fitted parameters are inspected [2], and extremely low learning rates are considered implausible. However, very difficult skills can have very low learning rates, and it is not clear what is the suitable threshold for defining low learning rates. Implausibility has been formally defined using model degeneracy [1], which refers to situations where parameter values violate the model's conceptual meaning. They defined strong empirical constraints to detect theoretical degeneracy, and designed two specific metrics involving empirical parameters to detect empirical degeneracy: (i) the model's estimated probability that a student knows a skill is not higher than before the student's first N actions, or (ii) the model doesn't assess that the student has mastered the skill, even though the student has made a large number M of correct responses in a row. Under these two cases, the model is judged to be empirically degenerate. They arbitrarily chose N=3 and M=10 for the study. A later theoretical fixed point analysis [19] has precisely identified the conditions where models will be empirically degenerate. We are interested in generally quantifying the plausibility property based on such a theoretical conclusion, avoiding imposing empirical parameters during evaluation.

Consistency. Prior work has focused on two aspects of this dimension. First, the optimization algorithm (namely, the Expectation Maximization algorithm) can converge to the local optima of the log likelihood space yielding different properties of parameters that depend on the initial values [5, 16]. Although there are studies on setting good initial values to tackle this problem [5], practically, the strategy of setting randomly distributed initial values is usually taken. Yet there is still no principled way to measure the models' difference in the variation of convergence, and as a result, it is difficult to get a quantitative view of such a property. Second, multiple global optima of Knowledge Tracing exist [16, 2] where observed student performance corresponds to different sets of parameter estimates that make different assertions about student knowledge, yet have identical (under finite precision) performance predictions [2]. This problem is referred to as the identifiability problem [2]. Later studies have presented different (and even contradictory) views of this problem [19, 9]. These two aspects all relate to the consistency of the parameter space, and in order to determine their practical implications, we offer a unified view of them.

3. POLYGON EVALUATION FRAMEWORK

Polygon is a novel framework proposed for evaluating general latent variable student models from multiple dimensions with multiple metrics, besides simply predictive performance. It considers three dimensions, predictive performance, plausibility, and consistency, along with novel metrics that instantiate each dimension. Polygon can evaluate a single model which contains only one set of parameters fitted from the data, because in practice we usually deploy a single model into a tutoring system after model selection. Polygon's predictive performance and plausibility metrics can be used to evaluate single models. However, latent variable models can converge to different points with different initial parameter values due to the non-convexity of the negative log-likelihood. A better model should be more likely to converge to points with higher predictive performance and plausibility, and also give more stable predictions and inferences. So we also use Polygon to evaluate a student model fitted from a large number of random initializations. This provides an examination on the parameter space that is useful for single model selection or construction. In our study, we call these final fitted models random restarts. We mainly focus on evaluating the parameter space from random restarts, but also include evaluating a single model. Each Polygon metric evaluates the trained model(s) of a skill. To get an overall evaluation across skills, we aggregate by averaging each skill's individual metric. All metrics range from 0 to 1, with a higher positive value indicating higher quality. We focus on the evaluation on Knowledge Tracing and FAST in this study. We now introduce Polygon in detail.

3.1 Predictive Performance

Predictive performance has been the previous standard of evaluating student models. It provides useful validation for the inference of knowledge, since accurate knowledge estimation should imply accurate prediction of student performance. We apply a widely used classification metric for this.

AUC and P-RAUC. We use Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve to evaluate each single model on test set, which gives an overall summary of diagnostic accuracy. AUC equals 0.5 for a random classifier and 1.0 for perfect accuracy. For assessing multiple random restarts, we compute the average of AUC values from single models and define it as P-RAUC, where P- stands for prediction performance, R stands for random restart, and r indicates the rth random restart:

$$\text{P-RAUC} = \frac{1}{R}\sum_{r=1}^{R}\text{AUC}^r \tag{1}$$

3.2 Plausibility

The conceptual idea behind using Knowledge Tracing to model student knowledge is that knowing a skill generally leads to correct performance, and conversely, that correct performance implies that a student knows the relevant skill [1]. We define plausibility metrics based on this idea.

Guess+Slip<1 (GS) and P-RGS. Several prior studies have empirically addressed the issue of plausibility, as mentioned in Section 2. A recent study [19] has provided a theoretical ground that we think can be used to formally define plausibility. This study used theoretical fixed point analysis to prove that when Guess+Slip>1, the probability that a student has learned a skill just after a practice, given the student's previous performance, decreases for correct practices and increases for incorrect practices. In this case, the model is empirically degenerate [1]. This is different from theoretically degenerate [1] constraining Guess≤0.5 and Slip≤0.5 to be plausible estimations, which we think is somewhat too strong. For example, it is possible that a student may answer a problem correctly after receiving strong scaffolding (help), even though the skill has not yet been learned. As a result, we propose a metric constructed using the Guess+Slip<1 condition. We use an indicator for Guess+Slip<1 for a single model and refer to it as GS (Equation 2). For assessing random restarts, we compute the average of the GS values from single models and define it as P-RGS, where P- stands for plausibility and R stands for random restart (Equation 3):

$$\text{GS}^r = \mathbb{1}(\text{Guess}^r + \text{Slip}^r < 1) \tag{2}$$

$$\text{P-RGS} = \frac{1}{R}\sum_{r=1}^{R}\text{GS}^r \tag{3}$$

Here, $\mathbb{1}$ is an indicator function and Guess^r and Slip^r are the r^{th} random restart's fitted probabilities. For FAST, with the change of feature values, Guess and Slip can change. We focus on capturing the average behavior of guessing and slipping across contexts, so we compute Guess and Slip with only the intercepts in the logistic regression component (note that other features are activated according to context during training). The interpretation of our computation depends on the construction of features. For example, when using item indicator features, the computation captures the average values of Guess and Slip of a skill.

Non-decreasing Predicted Probability of Learned (NPL) and P-RNPL. In addition to the above metric grounded in a theoretical analysis [19] for Knowledge Tracing, we construct another empirical metric to capture the behavior of a general latent variable model, since it is not always easy or feasible to conduct theoretical analysis of complex models. Our proposed metric captures how likely a model gives a non-decreasing estimation of knowledge levels with an increase in practice opportunities. This idea is consistent with constraining the learning rate to be non-negative, as in [17, 6]. We think that a decreasing predicted probability of learned is not plausible, based on the interpretation that such a decrease implies practices that hurt learning. We are aware that a decreasing knowledge estimate can also be interpreted as a decrease in the model's belief that a student might reach a high knowledge level, where the model adjusts itself when observing a lot of incorrect practices. However, we focus on the first interpretation, because in real world tutoring systems where students are aware of their knowledge level as provided by the systems, decreasing knowledge estimates with more practices might discourage students from trying more.

To construct this new metric, we first obtain the estimation of a student reaching leaned state at each t^{th} practice opportunity given prior 1^{th} to $(t-1)^{th}$ performance O_1 to O_{t-1} on the test set. We denote this probability as $\text{P}(\text{L}_t\text{=Learned}|\mathbf{O}_{1:t-1})$, and also refer to it as $\text{P}(\widetilde{L}_t|\mathbf{O})$ for simplicity. Then we count the total number of consecutive pairs with non-decreasing $\text{P}(\widetilde{L}_t|\mathbf{O})$ across each skill-student sequence, and then divide it by the the total number of observations of the current skill. We define this as NPL as an indicator of its plausibility for assessing a single model (Equation 4). For assessing random restarts, we compute the average of the NPL values obtained from single models, and define it as P-RNPL, where P- stands for plausibility and R stands for random restart (Equation 5):

$$\text{NPL}^r = \frac{1}{D}\sum_{s=1}^{S}\sum_{t=1}^{T_s-1} \mathbb{1}[\text{P}(\widetilde{L}_{t+1}^{rs}|\mathbf{O}^{rs}) \geq \text{P}(\widetilde{L}_t^{rs}|\mathbf{O}^{rs})] \quad (4)$$

$$\text{P-RNPL} = \frac{1}{R}\sum_{r=1}^{R}\text{NPL}^r \quad (5)$$

where $\mathbb{1}$ is an indicator function, r, s, t indicates random restarts, students, and practice opportunities, respectively. T_s is the total number of practices of student s, and D is the total number of practices of all students of current skill.

3.3 Consistency

Depending on different initial values of parameters, Knowledge Tracing and FAST can converge to points with different properties (such as plausibility or prediction of mastery). We favor a consistent model that has a low variance in properties across random restarts. Here, we extend the problem of Identifiability, where only global optimal log likelihood points are involved, into a more general problem of consistency, where all converged points are examined. The measurement of all converged points might be more operational in practice since it can be hard to judge whether the algorithm reaches a local or global optimum. For example, it is not clear how many random restarts are needed. Also, it is not sure whether converged points with log likelihood very close to the identified highest one can be treated as global optima or not.

Consistency of AUC, GS, NPL (C-RAUC, C-RGS, C-RNPL). Based on the explained importance of the performance metric AUC and the plausibility metrics GS and NPL, we think that a good model should also present low variance in these metrics across random restarts. As a result, we define consistency metrics C-RAUC, C-RGS, C-RNPL correspondingly by computing the standard deviation[1] of each single model's metrics across multiple random restart runs (r) on the test set with some transformation to map them into [0, 1] interval. Here, C- stands for consistency and R stands for random restarts. For example, for computing C-RAUC, we use the following formula:

$$\text{C-RAUC} = 1 - \sqrt{\frac{1}{R}\sum_{r=1}^{R}(\text{AUC}^r - \overline{\text{AUC}})^2} \quad (6)$$

Consistency of the Predicted Probability of Mastery (C-RPM). Student models are usually used to assess whether and when students reach mastery, based on which tutoring systems give adaptive instruction. A model lacking consistency in mastery prediction will lead to varying decision in instruction, which can have a significant impact on students. So we also construct a metric to quantify this consistency, inspired by previous studies [2, 15, 7]. We use the conventional definition of Mastery as the probability of Learned reaching 0.95 [4]. We compute P(L_t=Learned|$\mathbf{O}_{1:t}$), the posterior knowledge estimation of being in the Learned state at t^{th} practice updated by 1^{st} to t^{th} practice observations $\mathbf{O}_{1:t}$. We also refer to it as P($\widetilde{L}_t|\mathbf{O}$) for simplicity. We then compute the probability of reaching Mastery as the percentage of students predicted to ever have P($\widetilde{L}_t|\mathbf{O}$) ≥ 0.95, which means achieving a 0.95 posterior knowledge estimation in a practice sequence for the current skill. We refer to this probability as P(Mastery) or PM (Equation 7). We then compute the standard deviation of P(Mastery) across different runs, transform it to map to [0, 1] interval, and refer to it as C-RPM where C- stands for consistency, R stands for random restarts (Equation 8):

$$\text{PM}^r = \frac{1}{S}\sum_{s=1}^{S}\mathbb{1}\{\text{P}(\widetilde{L}_t^{rs}|\mathbf{O}^{rs}) \geq 0.95, \exists t \in [1, T_s]\} \quad (7)$$

$$\text{C-RPM} = 1 - \sqrt{\frac{1}{R}\sum_{r=1}^{R}(\text{PM}^r - \overline{\text{PM}})^2} \quad (8)$$

where r, s, t indicates random restarts, students, and practice opportunities respectively. T_s is the total number of practices of student s of current skill.

Cohesion of the parameter vector space (C-RPV). Fixed point analysis has been used to show that we need all four parameters to define the overall behavior of Knowledge Tracing [19] during the prediction phase, when knowledge estimation is updated by prior observations. We use this conclusion to construct another consistency metric. To capture all four parameters, we construct a Euclidian vector based on the four fitted parameters Init, Learn, Guess, and Slip for each single model. For FAST, we compute the four parameters with only the intercepts in the logistic regression components after fitting with features during training. We then compute the Euclidian distance of each vector to the mean of the parameter vectors (similar to the cluster cohesion measurement), and then perform a transformation to map this value to [0, 1] interval. We define it as C-RPV where C- stands for consistency, R stands for random restarts, and PV stands for parameter vector:

$$\text{C-RPV} = 1 - \frac{1}{2R}\sum_{r=1}^{R}||\mathbf{V}^r - \overline{\mathbf{V}}|| \quad (9)$$

where $\mathbf{V}^r$ is the parameter vector of the r^{th} random restart. $\mathbf{V}^r$=(Init^r, Learn^r, Guess^r, Slip^r). $\overline{\mathbf{V}}$ is the mean of the parameter vectors across the random restarts.

3.4 Metric Selection

Our proposed Polygon framework consists of three dimensions: prediction, plausibility, and consistency, and allows flexibly designed metrics for each dimension. The metrics we introduced before are the potential ones to be considered. We propose a principled way to select metrics to instantiate the framework: selected metrics should cover all three dimensions while having the smallest pairwise correlation. To achieve this, we examine the scatterplot and correlation of each pair of the metrics and conduct a significance test. Finally, we report our selected metrics in Section 4.3.1.

4. STUDIES AND RESULTS

4.1 Datasets and Features

We conducted experiments on datasets from different tutoring systems: Geometry Cognitive Tutor [12], OLI Engineering Statics [18], Java programming tutor [10], and the

[1] We use uncorrected sample standard deviations to map the metric to [0, 1]. With a large enough sample size (100 in our study), the bias of this estimator is small. For a smaller sample size, the corrected version might be considered.

Physics tutoring instance of the BBN learning platform [14]. Table 1 shows descriptive statistics (#observations indicates the smallest assessable practice units of students).

Geometry, Statics. We obtained these datasets from PSLC Datashop [13]. The Geometry dataset has data from the area unit of the Geometry course, which was conducted during the 1996-1997 school year. The Statics dataset has data from multiple schools during Fall 2011. We defined a problem (item) by concatenating the problem hierarchy, problem name, and step name. We defined a skill by concatenating the problem hierarchy and original skills, and treated the combination of skills as one unique skill if multiple skills are associated with a problem. For the Statics dataset, we randomly selected 20 skills (from the total of 156) to avoid bias towards this dataset when we aggregate across datasets. We further removed 3 skills where there are fewer than 10 observations in total, resulting in 17 skills. For FAST models, we constructed binary item indicator features for each problem with fitted coefficients represent item difficulties. Such models have been known for their high predictive performance [11, 8], and we plan to examine other dimensions as well.

Java. The Java dataset was collected from an online Java programming tutoring system [10] from Fall 2010 to Fall 2014. For each problem, students are asked to give the value of a variable or the printed output of a Java program after they have executed the code in their mind, and the system assesses correctness. The Java programs are instantiated randomly from templates on every attempt. Students can make multiple attempts until they think they have mastered the skill, or just give up. Problems are grouped by Java topics (each problem is mapped to a single topic), and we considered each topic as a skill. We consider each problem template as a single item. For FAST models, we also constructed binary item indicator features, adding to the exploration of the effect of item difficulties.

Physics. The Physics dataset was collected from the BBN Learning Platform [14], a domain-independent, problem-solving-based online learning platform. Students can solve problems without any help, or request a decomposition of the problem into steps. The steps lead students through a carefully crafted directed path to help solve the problem. We used logs collected from 40 users solving 10 problems from the Electric Circuits units. Each of these problems and steps are annotated with electric circuits skills (in total 10). In addition to capturing student actions at the items, the platform logs requests for help, feedback received, and problem navigation actions. We derived 105 numeric features from these logs, performed feature selection, and finally used the top ranked feature for FAST. This allows us to inspect the effect of help in the Knowledge Tracing framework.

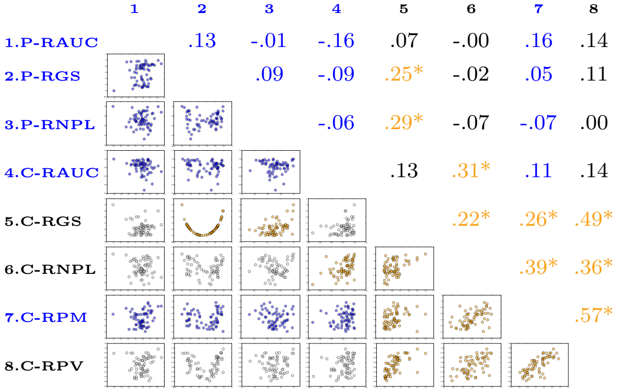
4.2 Experimental Setup

We used Expectation Maximization (EM) for training Knowledge Tracing, and Expectation Maximization with features for FAST [8]. We uniformly initialized each parameter within (0, 1) at each run for Knowledge Tracing, and we uniformly initialized each feature coefficient within (-10, 10) for FAST, which resulted in original parameters approximately covering (0, 1). We drew 100 different initial values for each parameter. We set 500 as the maximum EM iteration, 50 as the maximum LBFGS iteration and the log likelihood's relative change within 10^{-6} as convergence criteria. We trained each skill independently and used a user-stratified data split: 80% of the students were randomly selected into the training set, and the remaining students were assigned to the test set. In this way, models can be generalized to unseen students.

Table 1: Dataset descriptive statistics.

Dataset	#observations	#skills	#students	%correct
Geometry	5,055	18	59	75%
Statics	23,390	17	326	77%
Java	43,696	20	328	67%
Physics	10,063	10	40	62%

Table 2: Scatterplot and Kendall rank correlation among metrics of all skills (65) from Knowledge Tracing. Metrics selected into Polygon are shown in blue. Values shown in blue indicate a low correlation, and values shown in YellowOrange with asterisks indicate statistical significance (α=0.05).



	1	2	3	4	5	6	7	8
1.P-RAUC		.13	-.01	-.16	.07	-.00	.16	.14
2.P-RGS			.09	-.09	.25*	-.02	.05	.11
3.P-RNPL				-.06	.29*	-.07	-.07	.00
4.C-RAUC					.13	.31*	.11	.14
5.C-RGS						.22*	.26*	.49*
6.C-RNPL							.39*	.36*
7.C-RPM								.57*
8.C-RPV								

4.3 Results

4.3.1 Metric Selection

In order to obtain a compact instantiation of the Polygon evaluation framework, we analyze the pairwise correlation among the proposed metrics on Knowledge Tracing models. For each skill we compute eight metrics based on 100 random restarts and analyze the relationship across skills. Table 2 shows that C-RGS, C-RNPL and C-RPV all include significant correlations with other metrics. Particularly, the scatterplot of P-RGS and C-RGS shows a U-shape; we think this finding is because the mean and standard deviation of Bernoulli-distributed variables (GS) have this property. Finally, we instantiate the **Polygon** framework with five metrics in our study: **P-RAUC**, **P-RGS**, **P-RNPL**, **C-RAUC** and **C-RPM**, where they cover three dimensions and have low, non-significant pairwise correlations.

4.3.2 Evaluation on Multiple Random Restarts

We now present how we use Polygon to evaluate multiple random restart models and single models on Knowledge Tracing and FAST. Figure 1 shows Polygon evaluation per dataset aggregated across skills. Overall, FAST mostly have Polygon areas covering that of Knowledge Tracing. Considering the variance across skills, FAST has significantly higher values in all five metrics (α=0.05, $p < 0.0001$ by Wilcoxon signed-rank test), suggesting that it might promise not only higher predictive performance, but also higher plausibility and consistency. One possibility is that the constructed features indirectly constrain the optimization algorithm to

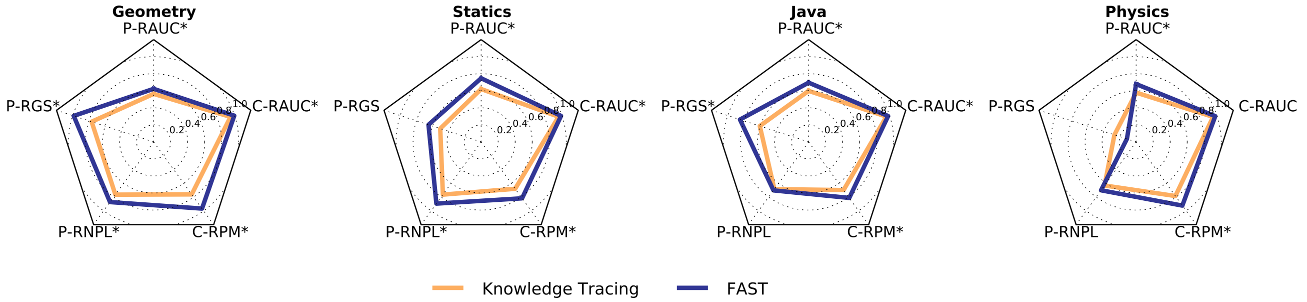


Figure 1: Polygon metrics per dataset comparing Knowledge Tracing and FAST. An asterisk (∗) indicates statistical significance under Wilcoxon signed-rank test (α=0.05). FAST's Polygon area mostly covers that of Knowledge Tracing.

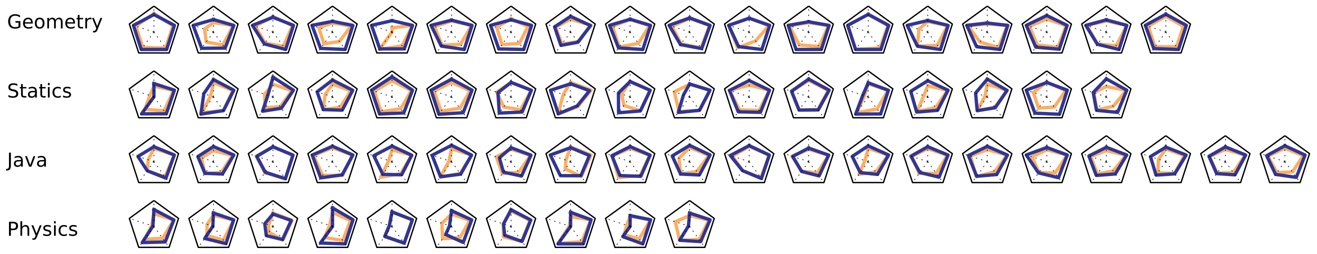


Figure 2: Polygon metrics per skill comparing Knowledge Tracing and FAST. FAST's Polygon area mostly covers that of Knowledge Tracing.

search within regions with both high fitness and plausibility. However, FAST's plausibility seems to be less stable, as compared to other properties, since its improvement varies across datasets.

We further examine Geometry, Statics and Java datasets where we use FAST with item difficulty features. As shown in Figure 1, FAST significantly outperforms Knowledge Tracing in all metrics, except for P-RGS on Statics and P-RNPL on Java, where FAST still presents positive tendencies. Generally speaking, using item difficulty features in Knowledge Tracing not only increases the model's predictive performance, but also its plausibility and consistency. However, the relative improvement in plausibility varies across datasets.

In the Physics dataset, FAST using problem decomposition requested features has a higher P-RAUC (significant), P-RNPL, C-RPM (significant), and C-RAUC, yet it also has a lower P-RGS, compared with Knowledge Tracing (not significant). Noticing that both methods have very low P-RGS, we suspect that skill definitions may be too coarse-grained, meaning that latter practices may involve potential new skills, where students fail more often than in the beginning. Thus, student models fitted from such data might be prone to estimating high Guess and Slip. FAST may be more vulnerable to bad skill definitions, since it might seek to fit the data as the primary goal, given that it has significantly higher predictive performance. In order to find out more about these potentially ill-defined skills, we further examine Polygon for each skill, as shown in Figure 2. This analysis shows that more than half of the skills in the Physics dataset have very low P-RGS, and particularly, there are two skills where FAST and Knowledge Tracing have an obvious gap on P-RGS (6^{th} and the last one), which should cause Knowledge Tracing to obtain a higher average value over FAST. We plan to examine whether refinement of the skill definitions will increase plausibility of both methods and FAST's relative quality for P-RGS in next steps.

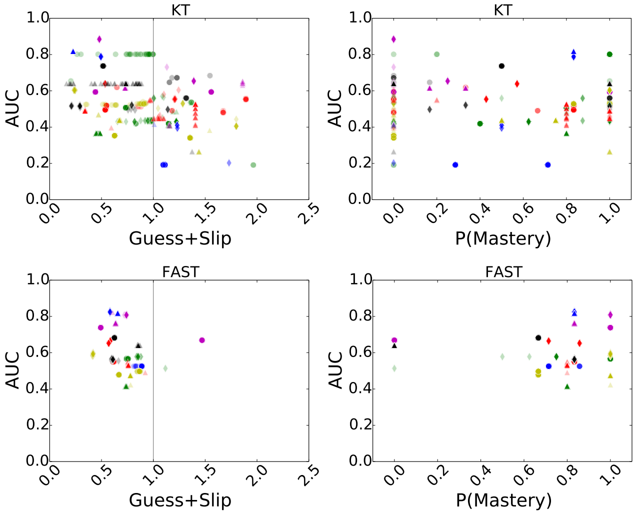


Figure 3: Evaluation on each skill's each random restart on Geometry dataset. Each color-shape corresponds to one skill. Each point corresponds to one random restart convergence point. Comparing with Knowledge Tracing, FAST generates more consistent, plausible models.

4.3.3 *Drill-down Evaluation of Single Models*

Polygon not only evaluates a method from multiple random restarts, but also contains components that can evaluate a single model. We use AUC, GS (Guess+Slip<1), and NPL to analyze each single model's predictive performance and plausibility, and also use the component PM (P(Mastery)) to get an intuitional understanding of a single model's effect on tutoring. Figure 3 visualizes AUC, Guess+Slip, and P(Mastery) of each random restart of each skill for Knowledge Tracing and FAST on Geometry dataset. Each color-shape corresponds to one skill, while each point corresponds to one random restart convergence point. We can easily determine different behaviors between Knowledge Tracing and FAST. FAST generates more consistent solutions than

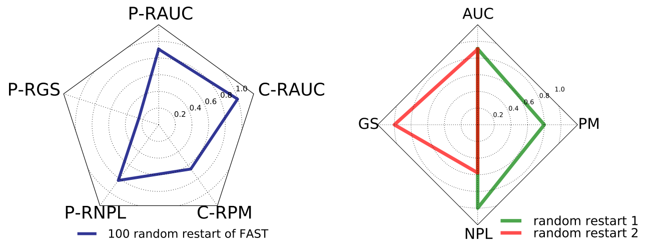


Figure 4: Polygon evaluation on a skill (id=154) on Statics dataset. The multi-model pentagon reveals this skill has high AUC consistency but low P(Mastey) consistency. The single-model quadrangle further reveals the contradictory properties of two random restart single models even they have very similar AUC.

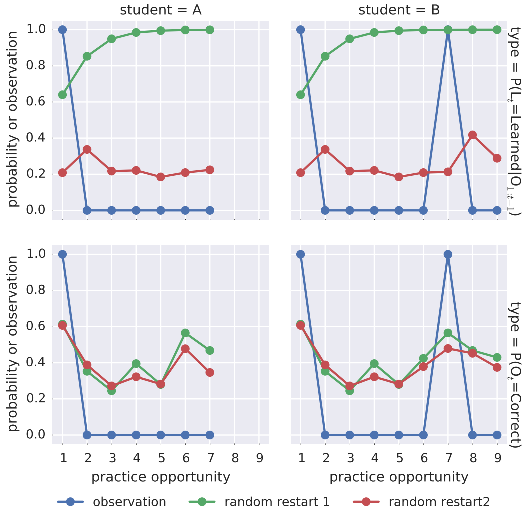


Figure 5: Comparison of two random restart single FAST models of a skill (id=154) from Statics dataset on two students. Both models have similar curves of predicted P(O$_t$=Correct) but have substantially different curves of predicted P(L$_t$=Learned | $\mathbf{O}_{1:t-1}$).

Knowledge Tracing, since there is less spread both horizontally and vertically of the random restart points within the same skill for all three metrics. FAST also generates more plausible models than Knowledge Tracing, since most of the points fall into Guess+Slip<1 region. Note that FAST asserts that students are more likely to reach mastery, since the converged points mostly lie in the higher-value region.

However, does FAST perform well on every skill? If not, can we use Polygon to effectively identify such skills and better understand the behavior? Based on previous skill-specific polygon evaluations (Figure 2), we identify one skill (3^{rd} polygon on the 2^{nd} row) on the Statics dataset, where Knowledge Tracing has better P-RGS than FAST. In Figure 4 the left-hand figure shows that this skill has a very high consistency of predictive performance (C-RAUC), yet a very low consistency of PM (C-RPM) across 100 random restarts. We further pick two of the random restarts and compute the polygon metrics for single models, as shown in Figure 4 right-hand single-model quadrangle. The quadrangle reveals that these two random restarts have almost identical AUC, yet have contradictory assertions about learning and mastery. In order to better understand the behavior, we pick two students from each one of these random restarts, and plot the predicted correctness curve and knowledge level curve (conditioned on prior observations). Figure 5 shows a severe problem in comparing these two random restarts: they have very similar predicted correctness, yet present fundamentally different predicted knowledge levels. We think that this problem extends the identifiability problem, in the sense that similar predicted correctness curves though not identical can be problematic if the predicted knowledge level curves differ greatly. Also, we observe the empirical degeneracy of random restart 1: with more incorrect practices, the predicted probability of Learned increases. This analysis showcases the deficiency of using only predictive performance to evaluate student models, and the effectiveness of Polygon metrics in identifying hidden problems.

Table 3: Kendall rank correlation among single model AUC, GS, NPL and log likelihood (LL) on training set for the same skill across 100 random restarts on Knowledge Tracing. We report the number of skills and in the bracket the average of the correlation values across skills under each positive (+) or negative (-) correlation relation (zero correlation ignored) among all skills (65).

	AUC		GS		NPL	
	+	−	+	−	+	−
AUC			41(0.6)	23(-0.6)	35(0.7)	30(-0.5)
LL	46(0.5)	19(-0.4)	34(0.5)	30(-0.5)	30(0.4)	35(-0.5)

4.3.4 *Implications for Single Model Selection*

We further examine the deficiency of using prediction performance or fitness metrics to select single models. We compute the Kendall rank correlation between AUC and the plausibility metrics for each run of each skill of Knowledge Tracing. Table 3 shows the deficiency of using only AUC to select the best random restart. There are more than one-third of skills that show a negative correlation between predictive performance and plausibility across different runs, and the magnitude of the negative correlation on average is not small. What about choosing the model with the maximum likelihood (LL) on the training set? Table 3 also shows the correlation between LL, AUC, and the plausibility metrics across different random restarts. Overall, about 71% (46/65) of the time, choosing the maximum LL on the training set can lead to a higher predictive performance in the test set, yet we have no more than 46% (30/65) of the time to get a more plausible model. These findings show that LL fails to offer a better choice than AUC. We think that a practical generalizable way to obtain a latent variable student model with both high predictive performance and plausibility remains to be explored, and Polygon provides important insights.

5. CONCLUSIONS

In this paper, we propose a general unified evaluation framework (that we call Polygon) to evaluate student models with latent knowledge estimates. Prior studies have presented different properties of the estimated parameters of Knowledge Tracing, yet there are no unified, quantitative evaluations for general student models. Our primary contribution lies in the quantitative unification of three aspects for general latent variable student models: predictive performance, plausibility, and consistency. We propose novel metrics and present a principled way to select proper metrics. Our defined dimensions extend the definitions of previously defined Identifiability and Model Degeneracy, which allows us to understand such problems more practically and more generally. A secondary contribution is that we show that a recent model with proper features, known as FAST, generally provides higher predictive models with higher plausibility and consistency than Knowledge Tracing. This suggests that proper features might help the optimization algorithm to constrain the search towards more plausible, more predictive regions.

There are several areas in which we can further extend our study. First, a single metric or perspective considering the multiple facets introduced in our analysis can further improve the accessibility of the evaluation. Also, each single metric can be further improved. For example, we can investigate the proper number of random restarts. However, Polygon's current individual metrics already provide insights for training student models. For example, incorporating the plausibility metric as a penalty into the optimization objective function can guide the algorithm to search within the high plausibility region. Second, external measurements applied in prior studies [4, 2, 6] may help to validate our framework. However, Polygon primarily serves as domain-independent internal validity, which is useful when external resources are not available. Third, the plausibility measurement can be a mixture of both student model and skill model evaluations. Will each model's relative quality be different when we examine well-defined vs. ill-defined skills? Can we utilize plausibility metrics to inspect skill model qualities? These are questions that remain unanswered. Fourth, we need to further understand and improve FAST. Since there are still cases where FAST generates models with low plausibility or low consistency, is there a principled way to construct features that maximize all three dimensions? Also, as we have only studied cases where a single feature (besides the intercept) is activated for each observation, will increasing the number of features change FAST's behavior?

Our study is still exploratory and serves as a first step towards a more theoretical, deeper understanding of the parameter estimation of complex latent variable student models. We hope that our work can open the door to more studies in the community on building student models that can yield not only better predictions of student performance but also more reliable, effective tutoring systems.

6. ACKNOWLEDGMENTS

This research is supported the Advanced Distributed Learning Initiative[2], Pearson[3] and the US Office of Naval Research (ONR) contract N00014-12-C-0535.

7. REFERENCES

[1] R. Baker, A. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems 2008*, pages 406–415. Springer.

[2] J. E. Beck and K.-m. Chang. Identifiability: A fundamental problem of student modeling. In *User Modeling 2007*, pages 137–146.

[3] T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. Painless unsupervised learning with features. In *HLT: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590.

[4] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[5] M. H. Falakmasir, Z. A. Pardos, G. J. Gordon, and P. Brusilovsky. A spectral learning approach to knowledge tracing. In *6th International Conference on Educational Data Mining*, pages 28–35, 2013.

[6] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent Tutoring Systems*, pages 35–44. Springer, 2010.

[7] J. P. González-Brenes and Y. Huang. Your model is predictive— but is it useful? theoretical and empirical considerations of a new paradigm for adaptive tutoring evaluation. In *Proceedings of the 8th Intl. Conf. on Educational Data Mining*, 2015.

[8] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *Proceedings of the 7th Intl. Conf. on Educational Data Mining*, 2014.

[9] G. Gweon, H.-S. Lee, C. Dorsey, R. Tinker, W. Finzer, and D. Damelin. Tracking student progress in a game-like learning environment with a monte carlo bayesian knowledge tracing model. In *Learning Analytics and Knowledge Conference 2015.*

[10] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky. Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. *Journal of Computer Assisted Learning, 2010.*

[11] M. Khajah, R. M. Wing, R. V. Lindsey, and M. C. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Proceedings of the 7th International Conference on Educational Data Mining*, 2014.

[12] K. R. Koedinger. Geometry area (1996-97), February 2014. In *URL https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=76.*

[13] K. R. Koedinger, R. S. J. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the EDM community: The PSLC DataShop. In *Handbook of Educational Data Mining*, pages 43–55, Boca Raton, FL, 2010. CRC Press.

[14] R. Kumar, G. Chung, A. Madni, and B. Roberts. First evaluation of the physics instantiation of a problem-solving based online learning platform. In *Intl. Conf. on Artificial Intelligence in Education 2015.*

[15] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Proceedings of the 5th Intl. Conf. on Educational Data Mining*, pages 118–125, 2012.

[16] Z. A. Pardos and N. T. Heffernan. Navigating the parameter space of bayesian knowledge tracing models: Visualizations of the convergence of the expectation maximization algorithm. *EDM*, 2010:161–170, 2010.

[17] P. Pavlik, H. Cen, and K. Koedinger. Performance Factors Analysis–A New Alternative to Knowledge Tracing. In *Proceeding of the 2009 conference on Artificial Intelligence in Education*, pages 531–538.

[18] P. Steif and N. Bier. Oli engineering statics - fall 2011, February 2014. . In *URL https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507.*

[19] B. van De Sande. Properties of the bayesian knowledge tracing model. *JEDM-Journal of Educational Data Mining*, 5(2):1–10, 2013.

[2] `http://www.adlnet.gov/`

[3] `http://researchnetwork.pearson.com/`