Your model is predictive— but is it useful? Theoretical and Empirical Considerations of a New Paradigm for Adaptive Tutoring Evaluation

José P. González-Brenes
Digital Data, Analytics and Adaptive Learning
Pearson School Research
Philadelphia, PA, USA
jose.gonzalez-brenes@pearson.com

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

ABSTRACT

Classification evaluation metrics are often used to evaluate adaptive tutoring systems— programs that teach and adapt to humans. Unfortunately, it is not clear how intuitive these metrics are for practitioners with little machine learning background. Moreover, our experiments suggest that existing convention for evaluating tutoring systems may lead to suboptimal decisions. We propose the Learner Effort-Outcomes Paradigm (Leopard), a new framework to evaluate adaptive tutoring. We introduce Teal and White, novel automatic metrics that apply Leopard and quantify the amount of effort required to achieve a learning outcome. Our experiments suggest that our metrics are a better alternative for evaluating adaptive tutoring.

Keywords

evaluation, efficacy, classification evaluation metrics

1. INTRODUCTION

A fundamental part of the scientific and engineering process is *testability*— the property of evaluating whether a hypothesis or method can be supported or falsified by data of actual experience. For example, in educational data mining, we formulate testable hypotheses that claim that the methods we engineer improve the outcomes of learners. In this manuscript, we study how to verify learner outcome hypotheses.

We focus on evaluating a popular type of educational method called *adaptive intelligent tutoring system.* Adaptive systems teach and adapt to humans; their promise is to improve education by optimizing the subset of *items* presented to students, according to their historical performance [5], and on features extracted from their activities [10]. In this context, items are questions, problems, or tasks that can be graded individually.

Evaluation metrics are important because they quantify the extent of whether an educational system helps learners. For example, a practitioner may use an evaluation method to choose which of the alternative adaptive tutoring systems to deploy in a classroom, or school district. On the other hand, a researcher may be interested in quantifying the improvements of her system compared to previous technology.

Our main contributions are proposing a novel evaluation paradigm for assessing adaptive tutoring and examples of when traditional evaluation techniques are misleading. This paper is organized as follows: § 2 reviews related methods for evaluating adaptive systems; § 3 describes the paradigm we propose for automatic evaluation of tutoring systems; § 4 provides a meta-evaluation of our novel evaluation techniques; and, § 5 provides some concluding remarks.

2. BACKGROUND

Adaptive tutoring is often implemented as a complex system with many components, such as a student model, content pool, and a cognitive model. Adaptive tutoring may be evaluated with randomized control trials. For example, in a seminal study [5] that focused on earlier adaptive tutors, a controlled trial measured the time students spent on tutoring and their performance on post-tests. The study reported that the tutoring system enabled significantly faster teaching, while students maintained the same or better performance on post-tests

Unfortunately, controlled trials can become extremely expensive and time consuming to conduct: they require institutional review board approvals, experimental design by an expert, recruiting (and often payment!) of enough participants to achieve statistical power, and data analysis. Automatic evaluation metrics improve the engineering process because they enable less expensive and faster comparisons between alternative systems. Fields that have agreed on automatic evaluation have seen an accelerated pace of technological progress. For example, the widespread adoption of the Bleu metric [15] in the machine translation community has lowered the cost of development and evaluation of translation systems. At the same time, it has enabled machine translation competitions that result in great advances of translation quality. Similarly, the Rouge metric [13] has helped the automatic summarization community transition from expensive user studies of human judgments that may take thousands of hours to conduct, to an automatic metric that can be computed very quickly.

The adaptive tutoring community has tacitly adopted conventions for evaluating tutoring systems [6, 16, 18]. Researchers often evaluate their models with classification evaluation metrics that assess the *student model* component of the tutoring system— student models are the subsystems that forecast whether a learner will answer the next item correctly. Popular classification evaluation metrics include accuracy, log-likelihood, Area Under the Curve (AUC) of the Receiver Operating Characteristic curve, and, strangely for classifiers, the Root Mean Square Error. However, automatic evaluation metrics are intended to measure an outcome of the end user. For example, the PARADISE [22] metric used in spoken dialogue systems correlates to user satisfaction scores. Not only is there no evidence that supports that classification metrics correlate with learning outcomes; but, prior work [2] has identified serious problems with them. For example, classification metrics ignore that an adaptive system may not help learners— which could happen with a student model with a flat or decreasing learning curve [1, 20]. A decreasing learning curve implies that student performance decreases with practice; this curve is usually interpreted as a modeling problem, because it operationalizes that learners are better off with no teaching. Therefore, an adaptive tutor with a student model with a decreasing learning curve does not teach students.

Surprisingly, in spite of all of the evidence against using classification evaluation metrics, their use is still very widespread in the adaptive literature [6, 16, 18]. Moreover, there is very little research on alternative evaluation techniques. A noticeable exception is recent work on individualizing student models [12]. The authors evaluated their approach using a method called *ExpOppNeed*, which calculates the expected number of practice opportunities that learners require to master the content of the tutoring curriculum. Though their evaluation methodology is extremely interesting and promising, it was not intended to be generalizable. In the next section we extend on prior work and present a novel general paradigm for evaluating adaptive systems.

3. LEOPARD EVALUATION

Adaptive tutoring implies making a trade-off between minimizing the amount of student *effort*, by carefully personalizing the curriculum, and maximizing student *outcomes* [4]. For example, repeated practice on a skill may improve student proficiency, at the cost of a missed opportunity for teaching new material. Adequate values for student effort and outcomes respond to external expectations from the social context. For example, it is not acceptable for a tutor to minimize effort by not teaching any content at all, or to maximize outcomes by taking twenty years to teach a simple concept. The right trade off is defined by subject matter experts.

We propose the novel Learner Effort-Outcomes Paradigm (Leopard) for automatic evaluation of adaptive tutoring. At its core, Leopard quantifies the effort and outcomes of students in adaptive tutoring. Even though measuring effort and outcomes is not novel by itself, our contribution is measuring both without a randomized control trial.

- Effort: Quantifies how much practice the adaptive tutor gives to students. In this paper we focus on counting the number of items assigned to students but, alternatively, amount of time could be considered.
- Outcome: Quantifies the performance of students after adaptive tutoring. For simplicity, we operationalize performance as the percentage of items that students are able to solve after tutoring. We assume that the performance on solving items is aligned to the long-term interest of learners.

We argue that Leopard is more intuitive than classification metrics because the effort and outcome resonate to educational principles. We now describe two novel metrics that apply the Leopard philosophy. In § 3.1, we describe Teal, a metric that calculates the theoretical expected behavior of students when interacting with a family of student models; and in § 3.2, we describe White[1] a metric that uses empirical data that may have not been collected on a control trial.

3.1 Theoretical Evaluation of Adaptive Learning Systems (Teal)

We formulate Theoretical Evaluation of Adaptive Learning Systems (Teal) to evaluate adaptive tutoring from the expected behavior of their student model. Teal focuses on models of the *Knowledge Tracing Family*— a very popular set of student models [10].

To use Teal on data collected from students, we first train a model using an algorithm from the Knowledge Tracing family (§ 3.1.1), then we use the learned parameters to calculate the effort (§ 3.1.2) and outcome (§ 3.1.3) for each skill. We discuss how to use Teal on models that use features (§ 3.1.4) and our design decisions (§ 3.1.5).

3.1.1 Knowledge Tracing Family

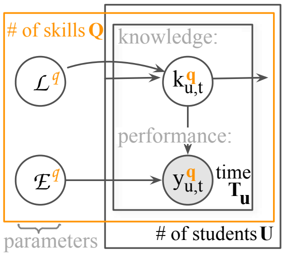


Figure 1: Knowledge Tracing plate diagram. The color of the circles represent whether the variable is latent (white), or observed in training (light), and plates represent repetition.

Figure 1 describes the Knowledge Tracing [5] model, the most simple member of the family. Knowledge Tracing requires a mapping of items to skills, often built by subject matter experts, although automatic approaches exist [8]. These skill mappings are also called cognitive models, or Q-matrices. Knowledge Tracing uses a Hidden Markov Model (HMM) per skill to model the student's knowledge as latent variables. The binary observation variable $y^q_{u,t}$ represents

[1] Tradition names metrics like colors! E.g., Rouge, Bleu.

whether the student u applies the t^{th} practice opportunity of skill q correctly. The latent variable $k^q_{u,t}$ models the latent student proficiency, which is often modeled with a binary variable to indicated mastery of the skill. To declutter notation, we may not explicitly write the indices q and u. There are two conventions for naming the skill-specific parameters of Knowledge Tracing. In the HMM tradition, the parameters are simply named transition or learning ($\mathcal{L}$), and emission ($\mathcal{E}$). In the educational tradition when using two latent states the parameters are called initial knowledge (ℓ_0), learning (ℓ), forgetting (f), guess (g) and slip (s). The Knowledge Tracing family includes models that parameterize the emission probabilities, transition probabilities, or both. For example, in Knowledge Tracing, the emission probability of emitting an answer $\mathbf{y}$ when the student has knowledge $\mathbf{k}$ is:

$$\mathcal{E}_{\mathbf{y},\mathbf{k}} = p(\mathbf{y}|\mathbf{k}) \tag{1}$$

Which is simply a binomial probability. To allow features in the emissions, we replace the binomial with a logistic regression [10]:

$$\begin{aligned}\mathcal{E}_{\mathbf{y},\mathbf{k}}(\boldsymbol{\beta}, \mathbf{X}_t) &= p(\mathbf{y}|\mathbf{k}; \boldsymbol{\beta}, \mathbf{X}_t) &(2)\\ &= \frac{1}{1 + \exp(-\boldsymbol{\beta}^{\intercal} \cdot \mathbf{X}_t)} &(3)\end{aligned}$$

Here $\mathbf{X}_t$ is the feature vector extracted at time t, and $\boldsymbol{\beta}$ is the regression coefficient vector. The feature may indicate, for example, if the student requested a hint.

3.1.2 Effort

Teal calculates the expected number of practice that an adaptive tutor gives to students. We assume a policy that the tutor stops teaching a skill once the student is very likely to answer the next item correctly according to a model from the Knowledge Tracing Family. For notational convenience, we define the probability of answering the next item correctly as:

$$c_{t+1}(\mathbf{y}_1, \ldots, \mathbf{y}_T) \equiv p(y_{t+1} = \text{correct}|\mathbf{y}_1, \ldots, \mathbf{y}_t; \mathcal{L}, \mathcal{E}) \tag{4}$$

Here $\mathcal{L}$ and $\mathcal{E}$ are the parameters of the Knowledge Tracing Family model. We can estimate c_{t+1} using conventional inference techniques for HMMs [19], such as the Forward-Backward algorithm.

The adaptive tutor teaches an additional item if two conditions hold: (i) it is likely that the student will get the next item wrong— in other words, the probability of answering correctly the next item is below a threshold R; and (ii) the tutor has not decided to stop instruction already. More formally, the tutor keeps teaching if:

$$\text{teach}(\mathbf{y}_1, \ldots, \mathbf{y}_t, R) \equiv \begin{cases} 1 & \text{if } \underset{t'<t}{\forall} c_{t'+1}(\mathbf{y}_1, \ldots, \mathbf{y}_{t'}) < R \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

We now can calculate at which practice opportunity the tutor should stop instruction. For simplicity, we assume all sequences are of length T. We simply count all of the times the tutor decides to teach a new item:

$$\text{cost}_R(\mathbf{y}_1, \ldots, \mathbf{y}_T) \equiv \sum_{t=1}^{T} \text{teach}(\mathbf{y}_1, \ldots, \mathbf{y}_t, R) \tag{6}$$

Note that if the probability of answering correctly the next item has not reached the threshold in T time steps, the cost is defined as T. Teal defines effort as the expected value of the number of practice opportunities a tutor gives. This is:

$$\begin{aligned}\text{effort}(R) &\equiv \mathbb{E}\left(\text{cost}_R(\mathcal{Y}_T)\right) &(7)\\ &= \sum_{\mathbf{y}_1,\ldots,\mathbf{y}_T \in \mathcal{Y}_T} \underbrace{\text{cost}_R(\mathbf{y}_1, \ldots, , \mathbf{y}_T)}_{\text{amount of practice}} \cdot \underbrace{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)}_{\text{sequence likelihood}} &(8)\end{aligned}$$

Here, $\mathcal{Y}_T$ is the set of all sequences of length T. When we have binary student outcomes (correct or not), the cardinality of this set is 2^T, which makes Teal only tractable for sequences of a few dozens of observations. In our experience, the sequences of adaptive tutoring systems are often in this range. In a companion paper [9] we give an alternative formulation of Teal that allows approximate calculations. The likelihood of the sequence can be efficiently estimated using the Forward-Backward algorithm.

3.1.3 Outcome

We define the outcome of a student as the mean performance after the tutor should stop instruction. For a particular sequence with student cost $k = \text{cost}_R(\mathbf{y}_1, \ldots, \mathbf{y}_T)$, this is:

$$\text{outcome}(\mathbf{y}_1, \ldots, \mathbf{y}_T, k) \equiv \begin{cases} \text{mean}(y_k \ldots y_T) & \text{if } k < T \\ \text{impute value} & \text{otherwise} \end{cases} \tag{9}$$

We map the correct and incorrect student responses y_t into 1 or 1, respectively. If the student sequence does not reach the performance threshold, we impute the value of the outcome. In this paper, we set the imputation value to 0. We define the score as the expected value of the outcome:

$$\begin{aligned}\text{score}(R) &\equiv \mathbb{E}(\text{outcome}(\mathcal{Y}_T, k)) &(10)\\ &= \sum_{\mathbf{y}_1,\ldots,\mathbf{y}_T \in \mathcal{Y}_T} \text{outcome}(\mathbf{y}_1, \ldots, \mathbf{y}_T, R) \cdot p(\mathbf{y}_1, \ldots, \mathbf{y}_T) &(11)\end{aligned}$$

3.1.4 Usage on Models With Features

For models that parameterize emission or transitions we first must build a counterfactual feature vector $\mathbf{X}$, and use it to calculate model parameters that do not depend on features. For example, consider a model that uses a binary feature vector that encodes students in different conditions. Conditions can be any feature of interest of the tutoring system, such as the ability to display multimedia content. We can use Teal to calculate the effort of students in each of the specific conditions.

For example, consider a feature vector $\mathbf{X} = (f_1, f_2, \ldots, f_n)$. Feature f_1 is 1 iff the student is using condition 1 (e.g., multimedia content is available), feature f_2 is 1, iff the student is using condition 2, etc. The vector is all zeros if the student is in the control condition. If we activate feature f_1, we can calculate the effort or score of students in the treatment 1. To apply Teal we first estimate counterfactual slip and guess parameters using Equation 3. We can use the counterfactual parameters with Teal.

For some models with features, Teal may require that students are assigned randomly to feature activation conditions, so that the regression coefficients can be interpreted as causal effects. Teal may not be appropriate if – for example – the features have reverse causality, or if there are omitted variables in the model.

3.1.5 *Design Discussion*

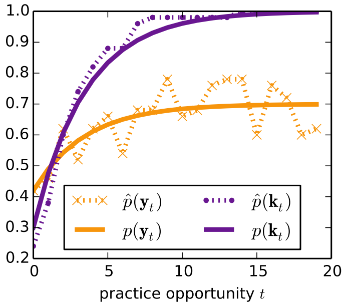


Figure 2: Expected and empirical student performance for a skill ($\ell_0 = 0.3$, $\ell = 0.25$, $g = 0.3$, $s = 0.3$, $f = 0$).

Teal extends the ExpOppNeed algorithm discussed on § 2. We compare both approaches to justify our design decisions.

1. **When to stop tutoring.** Teal expects tutoring to stop once the student is very likely to apply the skill correctly. On the other hand, ExpOppNeed relies on stopping tutoring once the posterior probability of the latent variable for knowledge is above a threshold. Figure 2 compares both approaches for some Knowledge Tracing parameters. The solid lines represent the expected values derived theoretically[2] for both strategies. To illustrate what actual student behavior may look like, we plotted dotted lines for 50 synthetic students sampled from a HMM. Although individual students vary, their average behavior is close to theoretical.

   In the figure, with 15 practice opportunities the students have close to 100% probability of skill mastery, while they only have 65% probability of applying the skill correctly. This big gap between the probability of mastery and probability of correct (the two solid lines) implies that the model is defining mastery as a state when students have low probability of applying the skill correctly. Low probability of answering correctly in a mastery state can occur due to a number of problems, for example, an incorrect item-to-skill mapping, or confusing tutoring content. We argue that an evaluation metric should penalize such models to be consistent with the Mastery Learning Theory [3].

   Moreover, prior work [1] has demonstrated that some ill-defined models have probability of correct decreasing with practice opportunities, at the same time that the probability of mastery increases. ExpOppNeed does not penalize such ill-defined models, but Teal does.

Algorithm 1 Single-Skill White

```
Require: performance sequences $\mathbf{y}_{u,q,t}$, student model predictions $\hat{\mathbf{c}}_{u,q,t}$ (the subscripts index students, skills, and practice opportunities), threshold $R$
function WHITE($\mathbf{y}_{u,q,t}, \hat{\mathbf{c}}_{u,q,t}, R$)
    for each student $u$ do
        for each skill $q$ do
            ▷ Select data for student $u$ and skill $q$ only:
            $\mathbf{y}', \hat{\mathbf{c}}' \leftarrow \text{filter}(\mathbf{y}, \hat{\mathbf{c}}, u, q)$
            $\text{effort}(q, u) \leftarrow 0$
            for each practice opportunity $t$ in $\mathbf{y}'$ do:
                if $\hat{\mathbf{c}}'_{t+1} \geq R$ then
                    $\text{score}(q, u) \leftarrow \text{mean}(y_{t+1}, \ldots, y_T)$
                    next skill $q$
                else if last($t$) then
                    $\text{score}(q, u) \leftarrow \text{impute}$
                $\text{effort}(q, u) \leftarrow \text{effort}(q, u) + 1$
    return effort, score
```

2. **What to measure.** ExpOppNeed does not calculate expected outcome of students. Teal considers both student outcome and effort because it is trivial to optimize one of the metrics if the other one is ignored.

3. **Precision of the results** Both ExpOppNeed and Teal have exponential computational complexity. However, ExpOppNeed uses a heuristic to prune sequences with low probability. Unfortunately, if the effort is very high (or infinite), the likelihood of the individual sequences becomes very low, and ExpOppNeed prunes the sequences too soon and therefore it may underestimate the effort. Teal improves on ExpOppNeed by defining effort on fixed-length sequences and not doing pruning.

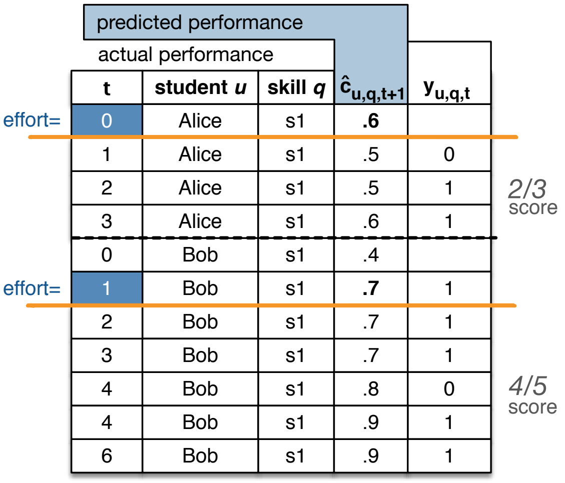
We now summarize some limitations of our approach. Teal assumes that the model parameters are correct, and does not take into account potential modeling problems— such as misspecification, or over-fitting. By design, Teal only is able to evaluate models in the Knowledge Tracing Family. We now present a novel evaluation method that addresses these limitations.

3.2 Whole Intelligent Tutoring System Empirical Evaluation (White)

We propose Whole Intelligent Tutoring System Evaluation (White), a novel automatic method that evaluates the recommendations of an adaptive system using data. White does not assume the student data is generated by a Knowledge Tracing model; instead, it relies on counterfactual simulations. White reproduces the decisions that the tutoring system *would* have made given the input data on the test set, by counting how many items the adaptive tutor would ask students to solve, and what is the mean student performance after tutoring.

Algorithm 1 describes White for a tutoring system that assumes an item is assigned to exactly one skill. We leave more complex tutors for future work. The input of White is the student performance sequences $\mathbf{y}$, the predictions of answering correctly $\hat{\mathbf{c}}$, and a threshold R that defines what is the

[2] Prior work derived [21]: $p(y_t = \text{correct}) = 1 - s - A\beta^t$. Here, $\beta = (1 - \ell)$, and $A = (1 - s - g) \cdot (1 - \ell_0)$


predicted performance
actual performance
effort=
effort=
2/3
score
4/5
score

t	student u	skill q	$\hat{c}_{u,q,t+1}$	$y_{u,q,t}$
0	Alice	s1	**.6**	
1	Alice	s1	.5	0
2	Alice	s1	.5	1
3	Alice	s1	.6	1
0	Bob	s1	.4	
1	Bob	s1	**.7**	1
2	Bob	s1	.7	1
3	Bob	s1	.7	1
4	Bob	s1	.8	0
4	Bob	s1	.9	1
6	Bob	s1	.9	1

Figure 3: Example of White calculating counterfactual score and effort using empirical data ($R = 0.6$).

target probability of correct. White assumes that the students are a random sample of the student population. The predictions are calculated by the student model component of the adaptive tutoring. For a data-driven student model, the predictions can be informed with the history preceding the current time step. For instance, to predict on the third time step, the student model may use the data up to the second time step. For example, for Knowledge Tracing:

$$\hat{c}_t = \hat{p}(y_t = \text{correct}|\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}) \tag{12}$$

Figure 3 shows example data of how White works for a 60% threshold ($R = 0.6$). For each student and skill in the test set, White estimates their counterfactual effort— how many items the student *would* have solved using the tutoring system. In our example, Alice does not get to practice the skill because the student model believes that she is likely to already know it (effort =0), but Bob is given one practice opportunity (effort=1). After Bob answers correctly the item, he is not given any more practice. White also calculates a counterfactual score to represent the student learning. It is the percentage of correct answers after the instruction would have stopped. The score is related to an existing classification evaluation metric called precision. Precision aggregates the entire dataset, while score is computed by students and skills. Although superficially it may sound as a small difference, our strategy allows us to avoid a special case of the Simpson's Paradox. In § 4.1.1 we discuss the issue more.

In this paper, when we report results with White, we impute the score of students that do not reach the threshold with their average performance. This is deliberately a different imputation strategy that we use with Teal, which assigns a score of zero to students that do not reach the threshold.

4. META-EVALUATION

In this section we meta-evaluate Leopard. We experiment with data from students (§ 4.1) and simulations (§ 4.2).

We compare these sets of metrics:

- **Conventional metrics**. We use classification evaluation metrics to evaluate how the student models predict future student performance. For this, we allow student models to use the history preceding the time step we want to predict.
- **Leopard metrics**. We use the score and effort as calculated by White and Teal. For simplicity we report the average scores across skills, and the sum of the mean effort. For U students and Q skills, this is:

$$\text{dataset score}(R) = \frac{1}{Q \cdot U} \sum_{q}^{Q} \sum_{u}^{U} \text{score}(q, u) \tag{13}$$

$$\text{dataset effort}(R) = \frac{1}{U} \sum_{q}^{Q} \sum_{u}^{U} \text{effort}(q, u) \tag{14}$$

4.1 Real Student Data

We use data collected from a commercial non-adaptive tutoring system for middle school Math. Our dataset includes only the first part of the entire curriculum, and contains students from the same grade from multiple schools. It contains approximately 1.2 million observations from 25,000 students. We randomly split the dataset into three sets of students. The training and test set have 60% and 20% of the students, respectively. The remainder of the data is reserved for future experiments not described in this paper. The item bank was mapped to skills in three different ways— the *coarse* definition maps the items into 27 skills, the *fine* definition into 90 skills, and the proprietary one is not reported.

4.1.1 Are predictive models always useful?

Assessing an evaluation metric with real student data is difficult because we often do not know the ground truth. To get around this, we now describe a strategy to select a subset of the dataset that we know the behavior of. Our main insight is that for adaptive tutoring to be able to optimize when to stop instruction, the student performance should increase with repeated practice (the learning curve should be increasing). Our strategy consists on selecting the subset of the data where student modeling may fail, because student performance remains flat or decreases with practice.

We first train a simplified Performance Factors Analysis [17] (PFA) model. We use a logistic regression for each skill:

$$p(y^q_{u,t}) = \frac{1}{1 + \exp(\boldsymbol{\beta}^q \cdot \mathbf{X}^q)} \tag{15}$$

The dimensions of $\mathbf{X}^q$ are the count of prior correct responses of the student and an intercept. We learn the parameters of the model $\boldsymbol{\beta}^q$ using constrained optimization— the regression coefficient for the effect of prior correct responses has to be non-negative.

We only use data from the skills that have zero regression coefficient for the effect of prior correct responses (flat or decreasing learning curve). Such skills are not suitable for an adaptive tutor because the PFA student model believes that practice does not influence student performance. More concretely, this PFA model would give infinite practice to difficult skills, or no practice to easy skills. Table 1 compares the results of using White and two conventional metrics on the test set of the selected skills. We compare with a majority class model that always predicts students answers as correct. The conventional metrics we report are the AUC, because of it's popularity, and the F-metric, because in experiments we report later correlates highly with White. For White we use a threshold of 60%. We cannot report on Teal because PFA is not part of the Knowledge Tracing Family.

Table 1: Evaluation metric comparison.

	White		conventional	
	score	effort	F	AUC
Performance Factors Analysis	.18	10.1	**.79**	**.85**
Majority Class	.18	11.2	0	.50

The AUC and F-metric results are arguably very high, indicating that the PFA model is highly predictive— yet by construction, we know that the model is *not useful* for adaptivity. The high prediction power of PFA is explained only by the intercepts of the model. That is, the predictions are based on the skill difficulty, independently of the student performance. We argue that White communicates better the unfavorable nature of the model because it reports a very low score, and only a small improvement of effort when compared to a baseline.

The problem with metrics that aggregate over the entire dataset, like the AUC and the F-metric, can be explained by Simpson's paradox— a trend that appears in different groups of data that disappears or reverses when the groups are combined. Because adaptive tutors learn a model from each skill independently, it is effectively a group of models. White and Teal evaluate each skill independently and are not susceptible to this problem. Consider the alternatives:

- Reporting as a baseline the *difficulty classifier*— a classifier that only considers the fraction of correct answers of each skill in the training set. For example, in Table 1, the PFA model has an AUC of 0.8, the same as the difficulty classifier. Because PFA did not outperform this baseline, it suggests the student model has a problem. However, simulations [8] provide evidence that useful student models may have predictive performance similar to the difficulty classifier. Therefore, the difficulty classifier baseline may reject some useful student models. Moreover, convention expects classifiers to have an AUC of higher than 0.5 to be useful, and this new baseline would break this interpretation.
- Calculating classification metrics over skills independently. This would only be useful when the skills are known beforehand, and not discovered with data [8]. We now provide evidence that suggests that classification metrics may be misleading, even when they are not affected by the Simpson's paradox.

4.1.2 Do traditional metrics lead to good decisions?

We now compare Leopard and traditional metrics for choosing an item-to-skill mapping. We train a PFA model using our Math dataset. Table 2 compares the results of White ($R = 0.6$) and AUC.

Table 2: Comparisons of item-to-skill definitions.

	White		
	score	effort	AUC
coarse	**.41**	**55.7**	.69
fine	.36	88.1	**.74**

If we were to choose the best skill mapping by AUC alone, we would choose the finer item-to-skill mapping, while White selects the coarser one. Why do they disagree? The fine skill mapping has almost three times the number of skills (90 skills) than the coarse mapping (27 skills). This means that for the effort to be the same on both models, the finer model should give a third of the practice of the coarser model. Even though the finer model is slightly more predictive, we argue that the coarser model is better suited for adaptive tutoring.

4.1.3 Case Study

For completeness, Table 3 demonstrates using different student modeling techniques with the coarse item-to-skill mapping. For Knowledge Tracing, we show both the White estimates, and the Teal estimates (in parenthesis). We use the average sequence length for each skill because Teal requires a sequence length as an input. The estimates of Teal and White for effort are very similar, but their scores mismatch— possibly due to the differences in imputation for skills that don't reach the threshold. The low score metrics are indicative of students not reaching the performance threshold. This suggests that further inspection is necessary, because the learning curves may be decreasing or some some skills may have high slip probabilities. One of the advantages of White is that it can be used to evaluate non-probabilistic student models. For example, we use White to evaluate the student model that gives practice of a skill until the student gets three correct answers in the skill.

Table 3: Student model comparison using Leopard

	Leopard		
	score	effort	AUC
Knowledge Tracing	.39 (.18)	**49.5** (50.9)	**.70**
Performance Factor Analysis	.41	55.7	.69
Three Correct	.39	59.1	n/a
Majority Class	.41	65.6	.50

4.2 Simulations

With real data, we do not know the extent that the parameters are learned correctly, or affected by modeling problems— such as misspecification. We now use synthetic data to evaluate different metrics and compare them to a ground truth. Given that we know the Knowledge Tracing parameters that were used to generate the synthetic datasets, we can use Teal to calculate *exactly* the student effort and outcomes.

We sample 500 different datasets using random Knowledge Tracing parameters. In none of the datasets we allow forgetting, but we do not impose any other constraint (not even that students improve with practice). Each dataset has only a single skill, and has 200 students with 10 practice opportunities. We do not learn parameters from the synthetic dataset, so we do not cross-validate.

4.2.1 *Which metrics correlate best with the truth?*

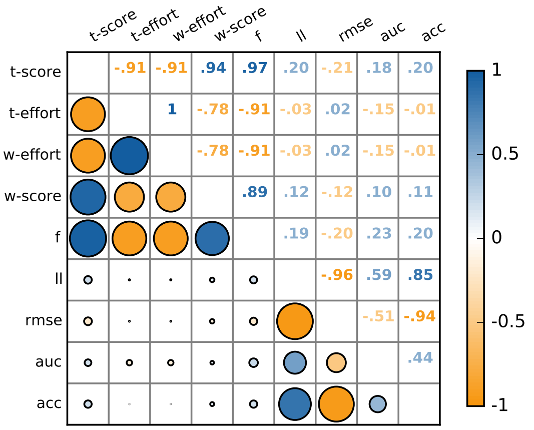


Figure 4: Correlation matrix of Leopard and conventional metrics. The size of the circles indicate the magnitude of the Pearson ρ correlation coefficient.

Figure 4 shows the pairwise Pearson-ρ correlations across 500 synthetic datasets on Teal (score), Teal (effort), White (effort), White (score), F-metric, Log-likelihood, RMSE, AUC, and Accuracy.

The metrics that correlate the most with the ground truth are White and the F-metric. Interestingly, the ground truth effort and score have low correlation with all the conventional metrics, except the F-metric, but the conventional metrics have relatively high correlation among each other (except the F-metric). In other words, most conventional metrics seem to be exchangeable.

We now investigate the effect of the imputation strategy of White. We are mindful that all of the synthetic students have 10 practice opportunities. Therefore, if White reports an effort of 10 for a dataset, it is likely that the dataset is not suitable for adaptivity, and that White may be imputing missing data to calculate the score. Figure 5 compares the 324 datasets that White reports effort lower than 9.99. Each dot in the scatterplot represents a different dataset. We see that effort computed with White has an almost perfect correlation with the ground truth ($\rho = 1.00$, p<0.05). On the other hand, the score computed with White is affected by our imputation strategy, but still has near perfect correlation ($\rho = 0.98$, p<0.05) with the ground truth. The correlation of the F-metric with the ground truth effort ($\rho = -0.47$) and score ($\rho = 0.89$) is relatively lower than White's. E.g., when the ground truth effort is 0, the F-metric ranges from very bad (0.2) to very good (1.0) predictive power, but White's effort is close to 0. Moreover, we speculate that score and effort may be more relatable to practitioners with little background of machine learning than the F-metric.

4.2.2 *Does White Converge to True Values?*

We now investigate whether White converges to the true values calculated by Teal. We use the same parameters used to plot Figure 2, and we manipulate the number of synthetic students, each student with 20 practice opportunities, Figure 6 shows that with little data, White converges to the true value computed by Teal. Future work may provide a formal argument of when and how much data White requires to convergence.

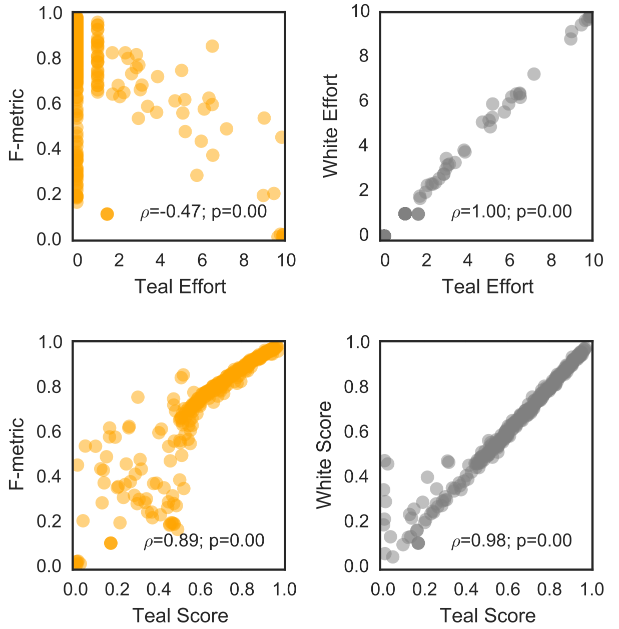


Figure 5: Comparison between F-metric and White to the ground truth.

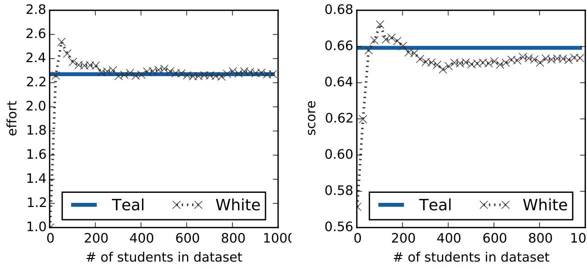


Figure 6: Example of White converging to Teal.

5. DISCUSSION

Our main contribution is the Leopard framework that automatically assesses adaptive tutoring systems in dimensions that relate to learner effort and outcomes. These dimensions were previously measured only in randomized control trials. We present Teal and White, two novel metrics that apply Leopard and are useful to evaluate adaptive tutoring systems. Secondary contributions include a novel methodology to assess evaluation metrics, the insight of Simpson's paradox affecting adaptive tutoring evaluation, and the implementation of the techniques we propose in this paper[3].

Classification evaluation metrics are very widespread in many disciplines, and their use in education is very important.

[3] `http://josepablogonzalez.com`

For example, for Computer-Adaptive Testing (CAT), classification metrics provide very useful insights to psychometric models. Leopard is not intended to replace classification metrics, randomized control trials, automatic experimentation [14], or visualization approaches [7, 11]. Leopard is a complementary approach to existing techniques, and we claim that it is specially useful when *in vivo* and online experimentation is not feasible.

We argue against the *de facto* standard of evaluating adaptive tutoring solely on classification metrics. Our experiments on real and synthetic data reveal that it is possible to have student models that are very predictive (as measured by traditional classification metrics), yet provide little to no value to the learner. Moreover, when we compare alternative tutoring systems with classification metrics, we discover that they may favor tutoring systems that require higher student effort with no evidence that students learn more. That is, when comparing two alternative systems, classification metrics may prefer a suboptimal system.

An interesting future direction may be to relax Teal's assumption that all sequences have fixed-length. Future work may provide more rigorous theoretical analysis on convergence, confidence intervals, validate our metrics with randomized control trials, or derive White for policies with multiple skills per item.

We are excited to see future work in adaptive tutoring systems reporting their contributions in terms of learner effort and outcomes. Besides the technical contributions of our evaluation metrics, we hope that our work contributes to the mission of driving the student modeling community to have a more learner-centric perspective.

6. 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 B. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, editors, *Intelligent Tutoring Systems*, volume 5091 of *Lecture Notes in Computer Science*, pages 406–415. Springer Berlin / Heidelberg, 2008.

[2] J. Beck and X. Xiong. Limits to accuracy: how well can we do at student modeling? In S. K. D'Mello, R. A. Calvo, and A. Olney, editors, *Proceedings of the 6th International Conference on Educational Data Mining, Memphis, Tennessee, USA, July 6-9, 2013*, pages 4–11. International Educational Data Mining Society, 2013.

[3] B. S. Bloom. Learning for mastery. *Evaluation Comment*, 1(2):1–12, 1968.

[4] H. Cen, K. R. Koedinger, and B. Junker. Is Over Practice Necessary?—Improving Learning Efficiency with the Cognitive Tutor Through Educational Data Mining. In *Proceedings of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 511–518, Amsterdam, The Netherlands, 2007. IOS Press.

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

[6] A. Dhanani, S. Y. Lee, P. Phothilimthana, and Z. Pardos. A comparison of error metrics for learning model parameters in bayesian knowledge tracing. Technical Report UCB/EECS-2014-131, EECS Department, University of California, Berkeley, May 2014.

[7] I. M. Goldin and A. Galyardt. Viz-r: Using recency to improve student and domain models. In *Proceedings of the 2nd ACM conference on Learning At Scale*, Vancouver, Canada, Mar. 2015.

[8] J. P. González-Brenes. Modeling Skill Acquisition Over Time with Sequence and Topic Modeling. In G. Lebanon and S. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics AISTATS 2015*, pages 296–305, 2015.

[9] J. P. González-Brenes and Y. Huang. Using data from real and simulated learners to evaluate adaptive tutoring systems. In *Proceedings of the Workshops at the 18th International Conference on Artificial Intelligence in Education AIED 2015*, Madrid, Spain, 2015.

[10] 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 M. Mavrikis and B. M. McLaren, editors, *Proceedings of the 7th International Conference on Educational Data Mining*, London, UK, 2014.

[11] Y. Huang, J. P. González-Brenes, R. Kumar, and P. Brusilovsky. A framework for multifaceted evaluation of student models. In J. G. Boticario, O. C. Santos, C. Romero, and M. Pechenizkiy, editors, *Proceedings of the 8th International Conference on Educational Data Mining*, Madrid, Spain, 2015.

[12] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In K. Yacef, O. R. Zaïane, A. Hershkovitz, M. Yudelson, and J. C. Stamper, editors, *Proceedings of the 5th International Conference on Educational Data Mining*, pages 118–125, Chania, Greece, 2012.

[13] C. Lin and E. Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51. Association for Computational Linguistics Morristown, NJ, USA, 2002.

[14] Y.-E. Liu, T. Mandel, E. Brunskill, and Z. Popović. Towards automatic experimentation of educational knowledge. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3349–3358. ACM, 2014.

[15] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics Morristown, NJ, USA, 2001.

[16] Z. A. Pardos and M. V. Yudelson. Towards moment of learning accuracy. In *Simulated Learners Workshop of Artificial Intelligence in Education*, 2013.

[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: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 531–538. IOS Press, 2009.

[18] R. Pelánek. A Brief Overview of Metrics for Evaluation of Student Models . In S. Gutierrez-Santos and O. C. Santos, editors, *Approaching Twenty Years of Knowledge Tracing Workshop of the 7th International Conference on Educational Data Mining*, London, UK, 2014.

[19] L. Rabiner and B. Juang. An introduction to Hidden Markov Models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

[20] D. Rai, Y. Gong, and J. E. Beck. Using dirichlet priors to improve model parameter plausibility. In T. Barnes, M. Desmarais, C. Romero, and S. Ventura, editors, *Proceedings of the 2nd International Conference on Educational Data Mining*, Cordoba, Spain, 2009.

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

[22] M. Walker, C. Kamm, and D. Litman. Towards developing general models of usability with PARADISE. *Natural Language Engineering*, 6(3):363–377, 2001.