Detecting the Moment of Learning

Ryan S.J.d. Baker[1], Adam B. Goldstein[2], Neil T. Heffernan[2]

[1] Department of Social Science and Policy Studies, Worcester Polytechnic Institute
100 Institute Road, Worcester MA 01609, USA
rsbaker@wpi.edu
[2] Department of Computer Science, Worcester Polytechnic Institute
100 Institute Road, Worcester MA 01609, USA
abgoldstein@gmail.com, nth@wpi.edu

Abstract. Intelligent tutors have become increasingly accurate at detecting whether a student knows a skill at a given time. However, these models do not tell us exactly at which point the skill was learned. In this paper, we present a machine-learned model that can assess the probability that a student learned a skill at a specific problem step (instead of at the next or previous problem step). Implications for knowledge tracing and potential uses in "discovery with models" educational data mining analyses are discussed, including analysis of which skills are learned gradually, and which are learned in "eureka" moments.

Keywords: Educational Data Mining, Bayesian Knowledge Tracing, Student Modeling, Intelligent Tutoring Systems

1 Introduction

In recent years, educational data mining and knowledge engineering methods have led to increasingly precise models of students' knowledge as they use intelligent tutoring systems. The first stage in this progression was the development of Bayes Nets and Bayesian frameworks that could infer the probability that a student knew a specific skill at a specific time from their pattern of correct responses and non-correct responses (e.g. errors and hint requests) up until that time [cf. 13, 18, 25].

In recent years, a second wave of knowledge modeling has emerged, which attempts to predict student knowledge more precisely based on information beyond just correctness. Beck et al [8] differentiated help requests from errors – however, doing so did not significantly improve predictive power. Baker, Corbett, & Aleven [3, 4] extended Bayesian Knowledge Tracing with contextualized estimation of the probability that the student guessed or slipped, leading to better prediction of future correctness. More recent work has suggested that the exact framework from [3, 4] leads to poorer prediction of post-test scores, but that information on contextual slip can be used in other fashions to predict post-test scores more precisely than existing methods [6]. Other knowledge tracing frameworks have attempted to model performance on problems or problem steps that involve multiple skills at the same time [cf. 21, 22], and have focused on predicting a student's speed of response in addition to just correctness [cf. 20].

Creating more precise models of student learning has several benefits. First of all, to the extent that student practice is assigned based on knowledge assessments [cf.

13], more precise knowledge models will result in better tailoring of practice to individual student needs [cf. 10]. Second, models of student knowledge have become an essential component in the development of models of student behavior within intelligent tutoring systems, forming key components of models of many constructs, including models of appropriate help use [1], gaming the system [5, 27], and off-task behavior [2, 11]. More precise knowledge models can form a more reliable component in these analyses, and reduce the noise in these models.

However, while these extensions to educational data mining have created the potential for more precise assessment of student knowledge at a specific time, these models do not tell us *when* the knowledge was acquired. In this paper, we will introduce a model that can infer the probability that a student learned a skill at a specific step during the problem-solving process. Note that this probability is ***not*** equal to P(T) in standard Bayesian Knowledge Tracing (a full explanation will be given later in this paper). Creating a model that can infer this probability will create the potential for new types of analyses of student learning, as well as making existing types of analyses easier to conduct. For example, this type of approach may allow us to study the differences between gradual learning (such as strengthening of a memory association [cf.20]) and learning given to "eureka" moments, where a skill is suddenly understood [cf. 17]. Do different skills lead to each type of learning?

To give another example, studying which items are most effective (and in which order they are most effective) [cf. 9, 23] will be facilitated with the addition of a concrete numerical measure of immediate learning. Similarly, studying the relationship between behavior and immediate learning is more straightforward with a concrete numerical measure of immediate learning. Prior methods for studying these relationships have required either looking only at the single next performance opportunity [cf. 12], a fairly coarse learning measure, or have required interpreting the difference between model parameters in Bayesian Knowledge Tracing [cf. 8], a non-trivial statistical task. Creating models of the moment of learning may even enable distinctions between behaviors associated with immediate learning and behaviors associated with learning later on, and enable identification of the antecedents of later learning. For example, perhaps some types of help lead to better learning, but the difference is only seen after additional practice has occurred.

In the following sections, we will present an approach for labeling data in terms of student immediate learning, a machine-learned model of student immediate learning (and indicators of goodness of fit), and an example of the type of "discovery with models" analysis that this type of model enables. In that analysis, we will investigate whether learning is differentially "spiky" between different skills, with learning occurring abruptly for some skills, and more gradually for other skills.

2 Data

The analyses discussed in this paper are conducted on data from 232 students' use of a Cognitive Tutor curriculum for middle school mathematics [16], during the 2002-2003 school year. All of the students were enrolled in mathematics classes in one middle school in the Pittsburgh suburbs which used Cognitive Tutors two days a week as part of their regular mathematics curriculum, year round. None of the classes were composed predominantly of gifted or special needs students. The students were in the 6th, 7th, and 8th grades (approximately 12-14 years old), but all used the same curriculum (it was an advanced curriculum for 6th graders, and a remedial curriculum for 8th graders).

Each of these students worked through a subset of 35 different lessons within the Cognitive Tutor curriculum, covering a diverse selection of material from the middle school mathematics curriculum. Middle school mathematics, in the United States, generally consists of a diverse collection of topics, and these students' work was representative of that diversity, including lessons on combinatorics, decimals, diagrams, 3D geometry, fraction division, function generation and solving, graph interpretation, probability, and proportional reasoning. These students made 581,785 transactions (either entering an answer or requesting a hint) on 171,987 problem steps covering 253 skills. 290,698 additional transactions were not included in either these totals or in our analyses, because they were not labeled with skills, information needed to apply Bayesian Knowledge Tracing.

3 Detecting the Moment of Learning

In this paper, we introduce a model that predicts the probability that a student has learned a specific skill at a specific problem step. We refer to this probability as P(***J***), short for "Just Learned". This model is developed using a procedure structurally similar to that in [3, 4], using a two-step process. First, predictions of student knowledge from standard Bayesian Knowledge Tracing are combined with data from future correctness and applications of Bayes' Theorem. This process generates labels of the probability that a student learned a skill at a specific problem step. Then a model is trained, using a broader feature set with absolutely no data from the future, to predict the labeled data.

3.1 Labeling Process

The first step of our process is to label each first student action on a step in the data set with the probability that the student learned the skill at that time, to serve as inputs to a machine learning algorithm. We label each student problem step (***N***) with the probability that the student learned the skill at that step. Specifically, our working definition of "learning at step ***N***" is learning the skill between the instant after the student enters their first answer for step ***N***, and the instant that the student enters their first answer for step ***N+1***.

We label step ***N*** using information about the probability the student knew the skill before answering on step ***N*** (from Bayesian Knowledge Tracing) and information on performance on the two following steps (***N+1***, ***N+2***). Using data from future actions gives information about the true probability that the student learned the skill during the actions at step ***N***. For instance, if the student probably did not know the skill at step ***N*** (according to Bayesian Knowledge Tracing), but the first attempts at steps ***N+1*** and ***N+2*** are correct, it is relatively likely that the student learned the skill at step

N. Correspondingly, if the first attempts to answer steps ***N+1*** and ***N+2*** are incorrect, it is relatively unlikely that the student learned the skill at step ***N.***

We assess the probability that the student learned the skill at step ***N,*** given information about the actions at steps ***N+1*** and ***N+2*** (which we term $\boldsymbol{A}_{+1+2}$), as:

$$\mathrm{P}(\boldsymbol{J}) = \mathrm{P}(\sim\boldsymbol{L_n} \wedge \boldsymbol{T} \mid \boldsymbol{A}_{+1+2})$$

Note that this probability is assessed as P($\sim\boldsymbol{L_n} \wedge \boldsymbol{T}$), the probability that the student did not know the skill and learned it, rather than P($\boldsymbol{T}$). Within Bayesian Knowledge Tracing, the semantic meaning of P($\boldsymbol{T}$) is actually P($\boldsymbol{T} \mid \sim\boldsymbol{L_n}$): P($\boldsymbol{T}$) is the probability that the skill will be learned, if it has not yet been learned. P($\boldsymbol{T}$)'s semantics, while highly relevant for some research questions [cf. 8, 16], are not an indicator of the probability that a skill was learned at a specific moment. This is because the probability that a student learned a skill at a specific step can be no higher than the probability that they do not currently know it. P($\boldsymbol{T}$), however, can have any value between 0 and 1 at any time. For low values of P($\boldsymbol{L_n}$), P($\boldsymbol{T}$) will approximate the probability that the student just learned the skill P($\boldsymbol{J}$), but for high values of P($\boldsymbol{L_n}$), P($\boldsymbol{T}$) can take on extremely high values even though the probability that the skill was learned at that moment is very low.

We can find P($\boldsymbol{J}$)'s value with a function using Bayes' Rule:

$$P(\sim\boldsymbol{L_n}\wedge\boldsymbol{T}|\,\boldsymbol{A_{+1+2}}) = \frac{P(\boldsymbol{A_{+1+2}}|\,\sim\boldsymbol{L_n}\wedge\,\boldsymbol{T}\,) * P(\sim\boldsymbol{L_n}\wedge\boldsymbol{T})}{P(\boldsymbol{A_{+1+2}})}$$

The base probability P($\sim\boldsymbol{L_n} \wedge \boldsymbol{T}$) can be computed fairly simply, using the student's current value for P($\sim\boldsymbol{L_n}$) from Bayesian Knowledge Tracing, and the Bayesian Knowledge Tracing model's value of P($\boldsymbol{T}$) for the current skill:

$$P(\sim\boldsymbol{L_n}\wedge\boldsymbol{T}) = P(\sim\boldsymbol{L_n})P(\boldsymbol{T})$$

The probability of the actions at time ***N+1*** and ***N***+2, P($\boldsymbol{A}_{+1+2}$), is computed as a function of the probability of the actions given each possible case (the skill was already known, the skill was unknown but was just learned, or the skill was unknown and was not learned), and the contingent probabilities of each of these cases.

$$\begin{aligned}P(\boldsymbol{A_{+1+2}}) = {} & P(\boldsymbol{A_{+1+2}} \mid \boldsymbol{L_n})\,P(\boldsymbol{L_n}) + P(\boldsymbol{A_{+1+2}}|\,\sim\boldsymbol{L_n}\wedge\boldsymbol{T})\,P\big(\sim\boldsymbol{L_n}\wedge\boldsymbol{T}\big)\\ & + P(\boldsymbol{A_{+1+2}}|\,\sim\boldsymbol{L_n}\wedge\sim\boldsymbol{T})\,P(\sim\boldsymbol{L_n}\wedge\sim\boldsymbol{T})\end{aligned}$$

The probability of the actions at time ***N+1*** and ***N***+2, in each of these three cases, is a function of the Bayesian Knowledge Tracing model's probabilities for guessing (***G***), slipping (***S***), and learning the skill (***T***). Correct answers are notated with a **C** and non-correct answers (e.g. errors or help requests) are notated with a **~C**.

$$\begin{array}{ll}
P(\boldsymbol{A_{+1+2}} = \boldsymbol{C},\boldsymbol{C}|\,\boldsymbol{L_n}) = P(\sim\boldsymbol{S})^2 & P(\boldsymbol{A_{+1+2}} = \boldsymbol{C},\sim\boldsymbol{C}|\,\boldsymbol{L_n}) = P(\boldsymbol{S})P(\sim\boldsymbol{S}) \\
P(\boldsymbol{A_{+1+2}} = \sim\boldsymbol{C},\boldsymbol{C}|\,\boldsymbol{L_n}) = P(\boldsymbol{S})P(\sim\boldsymbol{S}) & P(\boldsymbol{A_{+1+2}} = \sim\boldsymbol{C},\sim\boldsymbol{C}|\,\boldsymbol{L_n}) = P(\boldsymbol{S})^2 \\
P(\boldsymbol{A_{+1+2}} = \boldsymbol{C},\boldsymbol{C}|\,\sim\boldsymbol{L_n}\wedge\boldsymbol{T}) = P(\sim\boldsymbol{S})^2 & P(\boldsymbol{A_{+1+2}} = \boldsymbol{C},\sim\boldsymbol{C}|\,\sim\boldsymbol{L_n}\wedge\boldsymbol{T}) = P(\boldsymbol{S})P(\sim\boldsymbol{S}) \\
P(\boldsymbol{A_{+1+2}} = \sim\boldsymbol{C},\boldsymbol{C}|\,\sim\boldsymbol{L_n}\wedge\boldsymbol{T}) = P(\boldsymbol{S})P(\sim\boldsymbol{S}) & P(\boldsymbol{A_{+1+2}} = \sim\boldsymbol{C},\sim\boldsymbol{C}|\,\sim\boldsymbol{L_n}\wedge\boldsymbol{T}) = P(\boldsymbol{S})^2 \\
P(\boldsymbol{A_{+1+2}} = \boldsymbol{C},\boldsymbol{C}|\,\sim\boldsymbol{L_n}\wedge\sim\boldsymbol{T}) = P(\boldsymbol{G})P(\sim\boldsymbol{T})P(\boldsymbol{G}) + P(\boldsymbol{G})P(\boldsymbol{T})P(\sim\boldsymbol{S}) &
\end{array}$$

$$P(\boldsymbol{A}_{\boldsymbol{+1+2}} = \boldsymbol{C}, \sim\boldsymbol{C} \mid \sim\boldsymbol{L}_{\boldsymbol{n}} \wedge \sim\boldsymbol{T}) = P(\boldsymbol{G})P(\sim\boldsymbol{T})P(\sim\boldsymbol{G}) + P(\boldsymbol{G})P(\boldsymbol{T})P(\boldsymbol{S})$$
$$P(\boldsymbol{A}_{\boldsymbol{+1+2}} = \sim\boldsymbol{C}, \boldsymbol{C} \mid \sim\boldsymbol{L}_{\boldsymbol{n}} \wedge \sim\boldsymbol{T}) = P(\sim\boldsymbol{G})P(\sim\boldsymbol{T})P(\boldsymbol{G}) + P(\sim\boldsymbol{G})P(\boldsymbol{T})P(\sim\boldsymbol{S})$$
$$P(\boldsymbol{A}_{\boldsymbol{+1+2}} = \sim\boldsymbol{C}, \sim\boldsymbol{C} \mid \sim\boldsymbol{L}_{\boldsymbol{n}} \wedge \sim\boldsymbol{T}) = P(\sim\boldsymbol{G})P(\sim\boldsymbol{T})P(\sim\boldsymbol{G}) + P(\sim\boldsymbol{G})P(\boldsymbol{T})P(\boldsymbol{S})$$

Once each action is labeled with estimates of the probability P($\boldsymbol{J}$) that the student learned the skill at that time, we use these labels to create machine-learned models that can accurately predict P($\boldsymbol{J}$) at run-time. The original labels of P($\boldsymbol{J}$) were developed using future knowledge, but the machine-learned models predict P($\boldsymbol{J}$) using only data about the action itself (no future data).

3.2 Features

For each problem step, we used a set of 25 features describing the first action on problem step $\boldsymbol{N}$. The features used in the final model are shown in Table 1. 23 of those features were previously distilled to use in the development of contextual models of guessing and slipping [cf. 3, 4]. These features had in turn been used in prior work to develop automated detectors of off-task behavior [2] and gaming the system [5].

The 24th and 25th features were used in prior models of gaming the system and off-task behavior, but not in prior contextual models of guessing and slipping. These features are the probability that the student knew the skill before the first attempt on action $\boldsymbol{N}$, P($\boldsymbol{L}_{\boldsymbol{n-1}}$), and the probability that the student knew the skill after the first attempt on action $\boldsymbol{N}$, P($\boldsymbol{L}_{\boldsymbol{n}}$). There are some arguments against including these features, as P(~$\boldsymbol{L}_{\boldsymbol{n}}$) is part of the construct being predicted, P($\sim\boldsymbol{L}_{\boldsymbol{n}} \wedge \boldsymbol{T}$). However, the goal of this model is to determine the probability of learning, moment-by-moment, and the students' current and previous knowledge levels, as assessed by Bayesian Knowledge Tracing, are useful information towards this goal. In addition, other parameters in the model will be more interpretable if these terms are included. Without these terms, it would be difficult to determine if a parameter was predicting $\boldsymbol{T}$ or $\sim\boldsymbol{L}_{\boldsymbol{n}}$. With these terms, we can have greater confidence that parameters are predictive of learning (not just whether the skill was previously unknown), because $\boldsymbol{L}_{\boldsymbol{n}}$ is already accounted for in the model. However, in accordance with potential validity concerns stemming from including P($\boldsymbol{L}_{\boldsymbol{n-1}}$) and P($\boldsymbol{L}_{\boldsymbol{n}}$) in the model, we will also present goodness-of-fit statistics from a model not including these features.

While it is possible that features tailored to researchers' intuitions of what sorts of behaviors ought to predict moment-to-moment learning might perform better than these re-used features, the repeated utility of these features in model after model suggests that these features capture constructs of general applicability. Nonetheless, it will be valuable to consider additional features in future models of P($\boldsymbol{J}$). An additional aspect to consider with regards to the features is which actions the features are distilled for. As these features involve the first action at problem step $\boldsymbol{N}$, they represent the student's behavior at the moment right before learning, more than the student's behavior exactly at the moment of learning (which takes place in our model after the first action of problem step $\boldsymbol{N}$ and before the first action of problem step $\boldsymbol{N+1}$, as previously discussed). As such, the model's features should perhaps be interpreted as representing immediate antecedents of the moment of learning, as opposed to the exact characteristics of the moment of learning itself. Despite this limitation of our feature set, the model is still accurate at identifying the moment of learning (as discussed below). However, extending the model with a new set of features relevant to subsequent actions on problem step ***N*** (e.g. the second action to the last action) may improve model accuracy and interpretability.

3.3 Machine Learning

Given the labels and the model features for each student action within the tutor, two machine-learned models of P(***J***) were developed. As discussed above, one model used all 25 features, the other model only used the 23 features from [3, 4]. Linear regression was conducted within RapidMiner [19]. All reported validation is batch 6-fold cross-validation, at the student level (e.g. detectors are trained on five groups of students and tested on a sixth group of students). By cross-validating at this level, we increase confidence that detectors will be accurate for new groups of students. Linear Regression was tried both on the original feature sets and on interaction terms of the features; slightly better cross-validated performance was obtained for the original feature sets, and therefore we will focus on the models obtained from this approach.

3.4 Results

The model with 25 features, shown in Table 1, achieved a correlation of 0.446 to the labels, within 6-fold student-level cross-validation. The model with only 23 features achieved a weaker correlation of 0.301.

We can compute the statistical significance of the difference in correlation in a way that accounts for the non-independence between students, by computing a test of the significance of the difference between two correlation coefficients for correlated

Table 1. The machine learned model of the probability of learning at a specific moment. In the unusual case where output values fall outside the range {0,1}, they are bounded to 0 or 1.

Feature	**P(*J*) =**
Answer is correct	- 0.0023
Answer is incorrect	+ 0.0023
Action is a help request	- 0.00391
Response is a string	+ 0.01213
Response is a number	+ 0.01139
Time taken (SD faster (-) / slower (+) than avg across all students)	+ 0.00018
Time taken in last 3 actions (SD off avg across all students)	+ 0.000077
Total number of times student has gotten this skill wrong total	- 0.000073
Number of times student requested help on this skill, divided by number of problems	- 0.00711
Number of times student made errors on this skill, divided by number of problems	+ 0.0013
Total time taken on this skill so far (across all problems), in seconds	+ 0.0000047
Number of last 5 actions which involved same interface element	- 0.00081
Number of last 8 actions that involved a help request	+ 0.00137
Number of last 5 actions that were wrong	+ 0.00080
At least 3 of last 5 actions involved same interface element & were wrong	- 0.037
Number of opportunities student has already had to use current skill	- 0.0000075
F24: The probability the student knew the skill, after the current action (L_n)	- 0.053
F25: The probability the student knew the skill, before the current action (L_{n-1})	+ 0.00424
Constant Term	+ 0.039

samples [cf. 14] for each student, and then aggregating across students using Stouffer's Z [23]. According to this test, the difference between the two models is highly statistically significant, Z=116.51, p<0.0001.

Although correlation was acceptable, one limitation of this model is that it tended to underestimate values of P(***J***) that were relatively high in the original labels. While these values remained higher than the rest of the data (hence the model's reasonable correlation to the labels), they were lower, in absolute terms, than the original labels. This problem could be addressed by weighting the (rarer) high values more heavily during model-fitting, although this approach would likely reduce overall correlation.

As with any multiple-parameter linear regression model (and most other model frameworks as well), interpretability of the meaning of any parameter in specific is not entirely straightforward. This is because every parameter must be considered in the context of all of the other parameters – often a feature's sign can flip based on the other parameters in the model. Hence, significant caution should be taken before attempting to interpret specific parameters as-is. It is worth noting that approaches that attempt to isolate specific single features [cf. 8] are significantly more interpretable than the internal aspects of a multiple parameter regression model such as this one. It is also worth remembering that these features apply to the first action of problem step ***N*** whereas the labels pertain to the student's learning between the first action of problem step ***N*** and the first action of problem step ***N+1***. Hence, the features of this model can be interpreted more as representing the antecedents of the moment of learning than as representing the moment of learning itself – though they do accurately predict the moment of learning.

One interesting aspect of this model (and the original labels) is that the overall chance of learning a skill on any single step is relatively low within this tutor. However, there are specific circumstances where learning is higher. Many of these circumstances correlate to time spent, and the student's degree of persistence in attempting to respond. Larger numbers of past errors appear to predict more current learning than larger numbers of past help requests, for instance. This result appears at a surface level to be in contrast to the findings from [8], but is potentially explained by the difference between learning from requesting help once – the grain-size studied in [8] – and learning from requesting the same help sequence many times. It may be that learning from errors [cf. 26] is facilitated by making more errors, but that learning from help does not benefit from reading the same help multiple times.

4 Studying the Spikiness of Student Learning

A key way that the model presented here can be scientifically useful is through its predictions, as components in other analyses. Machine-learned models of gaming the system, off-task behavior, and contextual slip have proven useful as components in many other analyses [cf. 2, 12, 27]. Models of the moment of student learning may turn out to be equally useful.

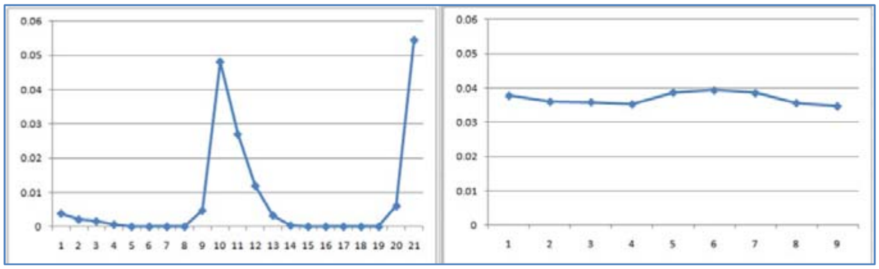

Fig. 1. A relatively “spiky” graph of a student’s performance on a specific skill, indicating eureka learning (left), and a relatively smooth graph, indicating more gradual learning (right). The X axis shows how many problem steps have involved the current skill, and the Y axis shows values of P(***J***).

One research area that models of the moment of student learning may shed light on is the differences between gradual learning (such as strengthening of a memory association [cf. 20]) and learning given to “eureka” moments, where a skill is understood suddenly [cf. 17]. Predictions of momentary learning for a specific student and skill can be plotted, and graphs which are “spiky” (e.g. which have sudden peaks of learning) can be distinguished from flatter graphs, which indicate more gradual learning. Examples of students experiencing gradual learning and eureka learning are shown in Figure 1. Note that the graph on the left in Figure 1 shows two spikes, rather than just one spike, a fairly common pattern in our data. Understanding why some spiky graphs have two spikes, and others have just one, will be an important area for future investigation. The degree to which learning involves a eureka moment can be quantified through a measure of “spikiness”, defined as the maximum value of P(***J***) for a student/skill pair, divided by the average value of P(***J***) for that same student/skill pair. This measure of spikiness is bounded between 1 (minimum spikiness) and positive infinity (maximum spikiness).

Spikiness may be influenced by the number of opportunities to practice a skill, as more opportunities may (by random variation) increase the potential maximum value of P(***J***). Therefore, to compare spikiness between skills, we only consider skills practiced at least 6 times, and only consider the first 20 steps relevant to that skill. Spikiness values range for skills between {1.12, 113.52}, M=8.55, SD=14.62. A valuable area of future work would be to study what characterizes the skills that have high spikiness and low spikiness. Spikiness values range for students between {2.22, 21.81}, M=6.81, SD=3.09, considerably less spikiness (on the whole) than the differences in spikiness seen between skills. Interestingly, however, a student’s spikiness is a good predictor of their final knowledge; the correlation between a student’s average final P($\boldsymbol{L_n}$) and their average spikiness is a very high 0.71, which is statistically significantly different than chance, F(1,228)=230.19, p<0.0001. This result suggests that learning spikes may be an early predictor of whether a student is going to achieve good learning of specific material.

5 Discussion and Conclusions

In this paper, we have presented a first model of P(***J***), the probability that a student learned a specific skill on a specific opportunity to practice and learn that skill. Though this model builds off of past attempts to contextualize student modeling [e.g. 3, 4] and to study the impact of different events on learning [e.g. 8, 23], this model is distinct from prior models of student learning, focusing on assessing the likelihood of learning on individual problem steps. We show that the model achieves acceptable correlation to the labels of this construct; there is still considerable room for improvement, potentially achievable through broadening the feature set.

We also show that the model's assessments of P(***J***) can be used to distill a secondary measure, the "spikiness" of learning, defined as the maximum momentary learning, divided by the average momentary learning. We find that a student's spikiness is an excellent predictor of their final knowledge, and that skills have greater variance in spikiness than students. Studying which aspects of skills predicts spikiness may be a valuable tool for further research into what types of skills are learned gradually or through "eureka" experiences. In addition, given the correlation between spikiness and final knowledge, models of P(***J***) are likely to prove useful for student knowledge modeling, as contextual guess and slip have been [e.g. 3, 4], and in the long term may lead to more effective adaptation by Intelligent Tutoring Systems.

Acknowledgements. This research was supported via NSF grant award number DGE0742503, by the Pittsburgh Science of Learning Center, NSF award number SBE-0836012, and by U.S. Department of Education program grant #R305A070440.

References

1. Aleven, V., McLaren, B., Roll, I., Koedinger, K.: Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. International Journal of Artificial Intelligence and Education, 16, 101-128 (2006).
2. Baker, R.S.J.d. Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems. In: Proc. ACM CHI: Computer-Human Interaction, 1059-1068 (2007).
3. Baker, R.S.J.d., Corbett, A.T., Aleven, V.: More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In: Proc. of the 9th Int'l. Conference on Intelligent Tutoring Systems, 406-415 (2008).
4. Baker, R.S.J.d., Corbett, A.T., Aleven, V.: Improving Contextual Models of Guessing and Slipping with a Truncated Training Set. In: Proceedings of the 1st International Conference on Educational Data Mining, 67-76 (2008).
5. Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R.: Developing a Generalizable Detector of When Students Game the System. User Modeling and User-Adapted Interaction, 18 (3), 287-314 (2008).
6. Baker, R.S.J.d., Corbett, A.T., et. al.: Contextual Slip and Prediction of Student Performance After Use of an Intelligent Tutor. Article under review.
7. Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., Koedinger, K.: Why Students Engage in "Gaming the System" Behavior in Interactive Learning Environments. Journal of Interactive Learning Research, 19 (2), 185-224 (2008).

8. Beck, J.E., Chang, K-m., Mostow, J., Corbett, A.: Does Help Help? Introducing the Bayesian Evaluation and Assessment Methodology. In: Proceedings of the 9th International Conference on Intelligent Tutoring Systems, 383-394 (2008).
9. Beck, J.E., Mostow, J. How who should practice: using learning decomposition to evaluate the efficacy of different types of practice for different types of students. In: Proc. of the 9th International Conference on Intelligent Tutoring Systems, 353-362 (2008).
10. Cen, H., Koedinger, K.R., Junker, B: Is Over Practice Necessary? Improving Learning Efficiency with the Cognitive Tutor. In: Proceedings of the 13th International Conference on Artificial Intelligence and Education.
11. Cetintas, S., Si, L., Xin, Y.P., Hord, C., Zhang, D.: Learning to Identify Students' Off-Task Behavior in Intelligent Tutoring Systems. In: Proceedings of the 14th International Conference on Artificial Intelligence in Education, 701-703 (2009).
12. Cocea, M., Hershkovitz, A., Baker, R.S.J.d.: The Impact of Off-task and Gaming Behaviors on Learning: Immediate or Aggregate? In: Proceedings of the 14th International Conference on Artificial Intelligence in Education, 507-514 (2009)
13. Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction, 4, 253-278 (1995).
14. Ferguson, G.A.: Statistical Analysis in Psychology and Education. New York: McGraw-Hill (1971).
15. Gong, Y., Rai, D., Beck, J., Heffernan, N.: Does Self-Discipline Impact Students' Knowledge and Learning? Proc. Of the 2nd International Conference on Educational Data Mining, 61-70 (2009).
16. Koedinger, K.R.: Toward evidence for instructional principles: Examples from Cognitive Tutor Math 6. In: Proceedings of PME-NA XXXIII (the North American Chapter of the International Group for the Psychology of Mathematics Education) (2002).
17. Lindstrom, P., Gulz, A.: Catching Eureka on the Fly. In: Proceedings of the AAAI 2008 Spring Symposium (2008).
18. Martin, J., VanLehn, K.: Student Assessment Using Bayesian Nets. International Journal of Human-Computer Studies, 42, 575-591 (1995).
19. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid Prototyping for Complex Data Mining Tasks. In: Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006), 935-940 (2006)
20. Pavlik, P.I., Anderson, J.R.: Using a Model to Compute the Optimal Schedule of Practice. Journal of Experimental Psychology: Applied, 14 (2), 101-117 (2008).
21. Pavlik, P.I., Cen, H., Koedinger, K.R.: Performance Factors Analysis – A New Alternative to Knowledge Tracing. In: Proceedings of the 14th International Conference on Artificial Intelligence in Education, 531-538 (2009).
22. Pardos, Z., Beck, J.E., Ruiz, C., Heffernan, N.T.: The Composition Effect: Conjunctive or Compensatory? An Analysis of Multi-Skill Math Questions in ITS. In: Proceedings of the 1st International Conference on Educational Data Mining, 147-156 (2008).
23. Pardos, Z., Heffernan, N.: Determining the Significance of Item Order in Randomized Problem Sets. In: Proc. of the 1st Int'l. Conf. on Educational Data Mining, 111-120 (2009).
24. Rosenthal, R., Rosnow, R.L.: Essentials of Behavioral Research: Methods and Data Analysis (2nd Edition). Boston, MA: McGraw-Hill (1991).
25. Shute, V.J.: SMART: Student modeling approach for responsive tutoring. User Modeling and User-Adapted Interaction, 5 (1), 1-44 (1995).
26. VanLehn, K., Siler, S., Murray, C., et. al.: Why Do Only Some Events Cause Learning During Human Tutoring. Cognition and Instruction, 21 (3), 209-249 (2003).
27. Walonoski, J.A., Heffernan, N.T.: Detection and analysis of off-task gaming behavior in intelligent tutoring systems. In: Proceedings of the 8th International Conference on Intelligent Tutoring Systems, 382-391 (2006).