

Avoiding Problem Selection Thrashing with Conjunctive Knowledge Tracing

K. R. KOEDINGER, P. I. PAVLIK JR., J. STAMPER,
Carnegie Mellon University, United States
T. NIXON, AND S. RITTER
Carnegie Learning Inc., United States

One function of a student model in tutoring systems is to select future tasks that will best meet student needs. If the inference procedure that updates the model is inaccurate, the system may select non-optimal tasks for enhancing students' learning. Poor selection may arise when the model assumes multiple knowledge components are required for a single correct student behavior. When the student makes an error, a deliberately simple model update procedure uniformly reduces the probability of all components even though just one may be to blame. Until now, we have had no evidence that this simple approach has any bad consequences for students. We present such evidence. We observed *problem selection thrashing* in analysis of log data from a tutor designed to adaptively fade (or reintroduce) instructional scaffolding based on student performance. We describe a *conjunctive knowledge tracing* approach, based on techniques from Bayesian networks and psychometrics, and show how it may alleviate thrashing. By applying this approach to the log data, we show that a third (441 of 1370) of the problems students were assigned may have been unnecessary.

Key Words and Phrases: Knowledge tracing, tutor log data analysis, Bayesian inference, blame assignment

1. INTRODUCTION

While educational data mining is often applied to discover patterns of students learning in data collected from instructional software, educational data mining can also be useful for identifying weaknesses in the tutoring systems that generated the data. This work presents an example of such identification revealed from analysis of the data and provides a detailed remedy based on Bayesian inference.

Student modeling depends on an accurate estimate of student knowledge to make effective instructional decisions. Making accurate inferences about what students know is challenging in situations where multiple knowledge components (skills, concepts, etc.) must be brought to bear, but where there is only one observation of student performance. If the student performs correctly, the credit assignment is straightforward. All the components get credit, because we have positive evidence that the student knows all the required components. However, if the student performs incorrectly, it is not necessarily appropriate to blame all the components. Any one or more of the components could be at fault. Determining which ones to blame is not straightforward. The Bayesian network [Millán et al. 2001] and psychometrics [Junker and Sijtsma 2001] literatures indicate how probability theory can be applied to address this problem. In this paper, we show how these ideas can be combined with Bayesian Knowledge Tracing [Corbett and Anderson 1995] to produce a “conjunctive knowledge tracing” approach.

Consider a simple example to illustrate the blame assignment problem. Imagine a tutor for teaching children to evaluate simple arithmetic expressions like “ $3*4+5$ ”. The student model could have knowledge components for each mathematical operator: addition, subtraction, multiplication, and division. The problem “ $3*4+5$ ” requires both multiplication and addition (we say “problem” here, but this argument applies more generally to any “step” in a problem solution that is performed as a separate observable action). If a student gets this problem step correct, we have evidence that they know both

Authors' addresses: K. R. Koedinger, Human Computer Interaction Institute, Carnegie Mellon U., Pittsburgh, PA., Email: koedinger@cmu.edu; P. Pavlik Jr, HCII, CMU, Email: ppavlik@andrew.cmu.edu; J. Stamper, HCII, CMU, Email: jstamper@cs.cmu.edu; T. Nixon, Carnegie Learning Inc., Pittsburgh, Email: tnixon@carnegielearning.com; S. Ritter, Carnegie Learning Inc. Email: sritter@carnegielearning.com.

the multiplication and addition components. If the student is incorrect, it could be that the student does not know multiplication and does not know addition, but it is also possible that the student knows addition but not multiplication or even multiplication but not addition. Consider the case where we have evidence from previous problems that the student is near mastery on addition, but has been struggling with multiplication. For example, the student has been successful on most problem steps that involve addition alone, like “14+3”, but has struggled on problems that involve multiplication alone, like “4*8”. In such a case, if a student makes an error on “3*4+5”, it is less likely to be a failure of addition and more likely a failure of multiplication. That is, the student is less likely to have been wrong because of not knowing addition and more likely to have been wrong because of not knowing multiplication.

In such a case, it does not seem appropriate to reduce the probability that the student knows addition as much as we would reduce the probability that the student knows multiplication. Nevertheless, equal blame assignment is simpler and was implemented as part of the original development kit for Cognitive Tutors [Corbett and Anderson 1995] and is currently used in practice in the widely distributed Carnegie Learning Cognitive Tutors [Ritter et al. 2007]. We pursue the problem of assigning blame in proportion to how likely it is that a knowledge component caused the error. Bayesian analysis provides a principled solution [cf. Millán et al. 2001, Junker and Sijtsma 2001].

We want a solution that not only works for two knowledge components (KCs) in combination, but one that generalizes to multiple KCs. For instance, in a harder problem step like $8-3*6$, the student model might have two more KCs like “following order of operations” and “dealing with negative numbers”. In this case, we want to distribute the blame appropriately across all four KCs depending on prior estimates of the KC difficulties. KCs with a higher prior probability of being known should receive less blame than KCs with lower probability. Pardos, Heffernan and Ruiz discuss this multiple-KC problem [Pardos et al. 2008]. Their proposed solution is to use additional diagnostic follow-up questions to determine the incorrect KC, and ignore the initial incorrect response to the question as a whole. Similarly, Cognitive Tutor interfaces are typically engineered so that correctness data on multiple individual steps in a problem solution strategy are available [Corbett and Anderson 1995]. However, in both approaches, the fine-grained diagnostic questions or steps (call them “scaffolds”) still sometimes have multiple KCs associated with them. Perhaps more importantly, in situations when this scaffolding is faded and a full question is given, neither approach provides an integrated diagnosis of the knowledge needed both for the relevant steps and for composing the steps together [Heffernan and Koedinger 1997]. A more elegant solution would be useful.

2. REVIEW OF KNOWLEDGE TRACING

Knowledge tracing is the student model update procedure used in Cognitive Tutors [Corbett and Anderson 1992]. For each knowledge component (KC), there is a two state hidden Markov model wherein there is a probability that the student is initially in either the known state (we use K_1 to represent this probability for “knowing” KC_1 or $Know-KC_1$) or the unknown state ($1-K_1$). There are three other parameters per KC: a slip probability (S) that a student will be incorrect even though they know the KC, a guess probability (G) that a student will be correct even though they do not know the KC, and a learning transition probability (T) that the student will learn at a particular tutoring opportunity and thus transition from the unknown to the known state. Because the challenge of the multiple-KC problem is in blame assignment, we only review here how the probability the student knows a KC is updated after an error observation (see Reye [1998] for a complete set of equations for knowledge tracing and related alternatives).

$$P(\text{Know-KC}_i|\text{Error}) = \frac{P(\text{Error}|\text{Know-KC}_i) * P(\text{Know-KC}_i)}{S * K_i + (1 - K_i) * (1 - G)} \quad (1)$$

The simplistic generalization of Equation 1 to the case where multiple KCs are involved on an incorrect step is to update each KC in the same way, that is, all required components are fully and equally blamed.

Table I. Example Consequences of Alternative Knowledge Tracing (KT) Approaches

Step	KC's Required			Knowledge Estimates			
	Add	Mult	Correct	Standard KT		Conjunctive KT	
				Add	Mult	Add	Mult
				0.960	0.300	0.960	0.300
3*4+5	1	1	0	0.700	0.270	0.955	0.297
6+3	1	0	1	0.938	0.270	0.993	0.297
7+4	1	0	1	0.990	0.270	0.999	0.297
4*7+3	1	1	0	0.893	0.267	0.999	0.287

Table I illustrates the results of standard knowledge tracing (see Standard KT columns) for a situation like the one described above. This simplified example is intended to clarify the process and consequences of the simplistic rule for blame, but, as we describe below, this example has the essential character of actual student data collected by an intelligent tutor in school use. The example assumes the student has mastered the knowledge component Add ($K_1 = .96$) but not Multiply ($K_2 = .3$). The probabilities of slipping, guessing, and learning parameters are set at 0.05, 0.2, and 0.25, respectively, for both KCs in this example. When a student makes an error on a problem step involving both Add and Multiply, like “3*4+5”, the estimates of knowing Add and Multiply are updated as follows. The estimate for Add (K_1) is updated according the formula above ($.05 * .96 / [.96 * .05 + (1 - .96) * (1 - .2)]$) to be 0.6. Knowledge tracing has a Markov property such that KCs have a probability of transitioning from the unknown state to the known state, that is, of being learned at each opportunity to learn. The transition probability in this example is 0.25 and when we apply it ($.6 + (1 - .6) * .25$) we get a new value for $K_1 = 0.7$. The analogous computations yield a new value for the Multiply, $K_2 = 0.27$.

The key point is that the Add KC drops significantly, to 0.70 – exactly as much as if the student had made an error on a problem step involving addition only (like 5+7). A sensible response of an intelligent tutor to this updated student model is to help the student get Add back up to mastery (a .95 threshold is used in Cognitive Tutors) by giving the student further practice (and as-needed instruction) on a problem involving Add (e.g., “6+3”). In fact, in this scenario, a student would have to get two problems involving Add right before getting back to mastery – see the 6+3 and 7+4 rows in Table I. The first raises the estimate to .938, still below a .95 mastery threshold, and the second to .990. If the student subsequently gets another problem with both KCs (e.g., “4*7+3”) wrong, the estimate for Add would again drop back below mastery. Another problem involving Add would then be selected. This would be wasting student time and energy if, in fact, they got the combined problem (“3*4+5”) wrong because of not knowing Multiply. In fact, the tutor and student might continue to thrash with the tutor repeatedly giving unneeded easy problems after the student errs on a harder problem.

Gong, Beck, & Heffernan [Gong et al. 2010] mentioned limitations of the knowledge-tracing algorithm when a problem or step is coded with multiple knowledge components. They were not addressing the issue, like we are, of on-line updates of the student model estimates of the probability a component has been learned. Others [Millán et al. 2001,

Junker and Sijtsma 2001] have presented relevant applications of Bayesian inference to address conjunctive combinations of skills and we build on that work.

3. CONJUNCTIVE KNOWLEDGE TRACING FOR FAIR BLAME ASSIGNMENT

The algorithm we present modifies knowledge tracing by changing the equations that deal with updating the student model after a student error (see Eq 1). The equations for updating after correct student responses are kept the same.

We present the case for two KCs first and generalize below to the case where multiple KCs are needed. Both the $P(\text{Error}|\text{Know-KC}_1)$ and $P(\text{Error})$ equations need to be modified. We use K_1 and K_2 to indicate the probabilities that KC_1 and KC_2 are known, S_1 and S_2 for their slip parameters, and G_1 and G_2 for their guess parameters. We start with $P(\text{Error})$, because it is simpler. An observed error can result from an unobserved error either in the execution of KC_1 or in the execution of KC_2 . An error in the execution of a KC occurs either when the KC is known but the student slips (e.g., $K_1 * S_1$) or when the KC is unknown and the student does not guess correctly (e.g., $(1-K_1)*(1-G_1)$). This formulation is shown in Equation 2.

$$P(\text{Error}) = \frac{K_1 S_1 + (1 - K_1)(1 - G_1) + K_2 S_2 + (1 - K_2)(1 - G_2)}{[K_1 S_1 + (1 - K_1)(1 - G_1)][K_2 S_2 + (1 - K_2)(1 - G_2)]} \quad (2)$$

We can find $P(\text{Error}|\text{Know-KC}_1)$ by plugging $K_1=1$ into the Equation 2 above and the result is shown in Equation 3.

$$P(\text{Error}|\text{Know-KC}_1) = \frac{S_1 + K_2 S_2 + (1 - K_2)(1 - G_2) - [S_1][K_2 S_2 + (1 - K_2)(1 - G_2)]}{S_1 + (1 - S_1)[K_2 S_2 + (1 - K_2)(1 - G_2)]} \quad (3)$$

An alternative formulation of Equation 2 that is easier to compute and easier to generalize to many KCs is shown in Equation 4.

$$P(\text{Error}) = 1 - P(\text{Correct}) \\ = 1 - [K_1(1 - S_1) + (1 - K_1)G_1][K_2(1 - S_2) + (1 - K_2)G_2] \quad (4)$$

Equation 4 computes the probability of error as one minus the probability of correct performance. To get a step correct requires that both KC_1 and KC_2 are executed correctly, which can be computed as the product of the probabilities of executing each KC correctly (this approach assumes KC execution is independent). Correct execution of a KC occurs either when the KC is known and the student does not slip (e.g., $K_1(1-S_1)$) or when the KC is unknown and the student guesses correctly (e.g., $(1-K_1)G_1$).

The combined update formula (Equation 5) gets applied for each KC, as was done in the example above. Applying this approach to the example above, we get the values shown in the ‘‘Conjunctive KC’’ columns in Table I. After the student made an error on ‘‘3*4+5’’, the estimate for Add (K_1) was updated according to the formulas above to 0.94.

$$\begin{aligned} & \frac{P(\text{Know-KC}_1|\text{Error})}{P(\text{Error}|\text{Know-KC}_1)} = \frac{P(\text{Know-KC}_1)}{P(\text{Know-KC}_1)} / \frac{P(\text{Error})}{P(\text{Error})} \\ & = \frac{.05 + (1 - .05)[.3 * .05 + (1 - .3)(1 - .2)]}{.96} / \frac{1 - [.96(1 - .05) + (1 - .96).2][.3(1 - .05) + (1 - .3).2]}{1} \end{aligned} \quad (5)$$

Applying the learning (or transition) probability ($.94 + (1-.94)*.25$) yields a new value for $K_1 = 0.955$. The analogous steps yield a new value for the Multiply, $K_2 = 0.297$. Unlike Standard Knowledge Tracing, the estimate for Add, at 0.955, stays above the mastery threshold of .95 and thus the tutor would not assign a potentially unnecessary addition problem. The potential is thus reduced for unproductive cycling back and forth or thrashing between hard and easy problems that may occur with standard knowledge tracing (as illustrated in Table I).

The key insight for blame assignment with two KCs is that the probability of being incorrect given that KC_1 is known is no longer just the probability of slipping on KC_1 .

There is also a chance that the student made an error in executing KC_2 . To generalize to multiple KCs, we need the $P(\text{Error}|\text{Know-KC}_1)$ formula to account for the possibility that an error can result from failure to execute on any of the other needed KCs.

First, Equation 6 shows the general equation for $P(\text{Error})$ when we use the $1 - P(\text{Correct})$ formulation (as anticipated in Equation 4) and compute $P(\text{Correct})$ as the product of executing all of the N KCs correctly:

$$\begin{aligned} P(\text{Error}) &= 1 - P(\text{Correct}) \\ &= 1 - \prod_{i=1}^N [K_i(1 - S_i) + (1 - K_i)G_i] \end{aligned} \quad (6)$$

Now, for the general equation of $P(\text{Error}|\text{Know-KC}_j)$ we need to a way to compute the disjunction (logical or) of executing incorrectly all of the required KCs besides KC_j . Because conjunctions are simpler to compute than disjunctions, we use the transformation in Equation 7 to formulate Equation 8.

$$\begin{aligned} P(\text{A or B or C}) &= \text{not}(\text{not } P(\text{A}) \text{ and not } P(\text{B}) \text{ and not } P(\text{C})) \\ &= 1 - [1 - P(\text{A})][1 - P(\text{B})][1 - P(\text{C})] \end{aligned} \quad (7)$$

Equation 8 replaces the term in Equation 3 for incorrect execution of K_2 with the disjunction of incorrect execution of all the required KCs but KC_j . Thus, note the use of “excluding KC_j ” in Equation 8. And note, as per Equation 7, the use “1-” both outside and inside the product (\prod).

$$\begin{aligned} P(\text{Error}|\text{Know-KC}_j) &= S_j + (1 - S_j) * (1 - \prod_{i \in \text{KCs, excluding } KC_j} [1 - [K_i(1 - S_i) + (1 - K_i)G_i]]) \end{aligned} \quad (8)$$

Finally, Equation 9 is the Conjunctive Knowledge Tracing alternative to blame assignment in Standard Knowledge Tracing (Equation 1) and it completes the generalization from two KCs (Equation 5) to any number of KCs.

$$\begin{aligned} P(\text{Know-KC}_1|\text{Error}) &= \frac{P(\text{Error}|\text{Know-KC}_1) * P(\text{Know-KC}_1)}{P(\text{Error})} \\ &= \frac{\text{Eq.8} * K_1}{\text{Eq.6}} \end{aligned} \quad (9)$$

4. CONJUNCTIVE KNOWLEDGE TRACING ON REAL DATA

In the introduction, we illustrated the possibility of a thrashing problem that can result from unfair blame assignment whereby a student is repeatedly assigned a hard problem (which they get wrong) and then unnecessary easy problems (which they tend to get right). We turn to a demonstration of this thrashing problem in real student use of a tutor. We then describe how use of Conjunctive Knowledge Tracing can alleviate this problem. The data come from 120 students working on a geometry area unit of the Bridge to Algebra Cognitive Tutor and, in particular, from an experiment to test a new KC model produced through a human-machine discovery method [Stamper and Koedinger 2011].

This implementation of the tutor used standard knowledge tracing, but we did make a change to the problem selection algorithm designed to create a better learning experience. The original problem selection tries to find problems that have the most opportunities for the student to address their least-mastered KCs (along with other factors, like minimizing the number of mastered KCs and encouraging variety). In the usual situation where there is only one KC per problem step this has been a reasonable approach. However, when there are multiple KCs per step, this current "maximize unmastered" algorithm criteria for problem selection will prefer problems that involve more unmastered KCs per step (harder problems) over problems that have fewer unmastered KCs per step (easier problems). In order to create a gentle slope in the learning trajectory, we modified the original problem selection algorithm to select problems that have as few unmastered KCs (but at least 1) as possible. Thus, students are more likely to be given easier (but not mastered) problems first and then, once these appear to be mastered, more complex

problems are selected. If, in turn, evidence from poor performance on complex problems suggests weaknesses in specific component KCs, easier problems will be selected again to bolster student mastery before returning to hard problems. The intention, then, is to adjust difficulty (fading or reintroducing scaffolding) to optimally adapt to student needs. This change revealed the thrashing problem and a practical weakness of standard knowledge tracing when multiple KCs are required on a step. The goals of the change in problem selection were to adaptively fade and “unfade” (reintroduce) scaffolding based on student performance. Fading occurs in transition from “scaffolded” problems, which tend to have 1 KCs per step, to “unscaffolded” problems, which tend to have key steps with multiple KCs. It is adaptive in that the transition occurs after students have demonstrated mastery of the KCs in the scaffolded problems. Scaffolding may be reintroduced based on evidence of too much failure on unscaffolded problems.

4.1. Results: Problem selection thrashing from poor blame assignment

Similar to the arithmetic example above, we modified a geometry area unit of the Bridge to Algebra Cognitive Tutor to include a mix of harder problem types in which some steps require many KCs (e.g., setting and executing subgoals to find a square area, circle area, and the difference) and easier problem types in which steps require just one or a few KCs (e.g., subtracting two areas). Four types of problems culminated with the student finding the area of an irregular shape (e.g., the left-over area when a circle is cut from a square) from the regular shapes that make it up. To aid understanding of the example of real student performance shown in Table II, we describe these problem types. The easiest problem type, called an “area scaffold problem” and displayed as Easy in Table II, gives the areas of the component shapes to focus students’ attention on how to combine them to find the irregular shape rather than on finding component areas themselves. The student need only recognize the need for area composition (the Comp KC in Table II) and perform the addition or subtraction (AddAreas and SubtrAreas KCs in Table II). The slightly less easy “table scaffold” problems (displayed as Easy’ in Table II) require the student to find the regular areas on their own, but explicitly prompt (or scaffold) the student to do so with a labeled column in a table interface widget where the areas are to be entered. While these problems require area computations (see the Area KC in Table II), those computations are separate steps in the interface and so the Area KC is not involved in the “composition” step to compute the irregular area that is displayed in Table II. In the harder “no scaffold” problems, students are asked to enter only the final irregular area (requiring up to four KCs in a single step) without any interface support to first find the component areas.

Turning to the student performance data, we found that the new problem selection algorithm described above worked well in that the easiest problem type (area scaffold) tended to be selected before the somewhat less easy problem type (table scaffold) and these before the hardest problem types (problem scaffold and no scaffold). However, we were surprised at how many of the easier problems students were given. On closer inspection we found the kind of cycling between easy and hard we illustrated above.

Table II provides an example from one of the students. The results are displayed starting after the student has been successful on two Easy problems and failed on a Hard problem. Before describing this example in more detail, first note how the student keeps getting assigned many Easy problems (and succeeds at them). These problems were assigned based on standard knowledge tracing (SKT), but, if conjunctive knowledge tracing (CKT) had been used, the five problems in the bolded row numbers (5, 8, 10, 12, and 14) would not have been assigned. In these rows, all of the CKT estimates are above 0.95 whereas some of the SKT estimates are not (see the bolded numbers). SKT assigns

these Easy problems because when errors are made on Hard problems, it attributes too much blame to easy KCs (SubtrAreas & AddAreas) that should be primarily attributed to hard KCs (SubGoal).

Table II. Problem selection thrashing from poor blame assignment in real student data.

Row	Prob Type	Corr	Standard KT Estimates					Conjunctive KT Estimates				
			Comp Areas	Subtr Areas	Add Areas	Area	Sub Goal	Comp Areas	Subtr Areas	Add Areas	Area	Sub Goal
1			0.98	1.00	0.62	0.62	0.49	0.98	1.00	0.86	0.86	0.70
2	Easy'	0	0.98	1.00				0.98	1.00			
3	Easy'	1	0.89	0.98				0.94	0.99			
4	Easy	1	0.98		0.62			0.99		0.86		
5	Easy	1	1.00		0.91			1.00		0.97		
6	Hard	0	1.00	1.00		1.00	0.49	1.00	1.00		1.00	0.70
7	Hard	0	1.00	0.98		1.00	0.38	1.00	1.00		1.00	0.68
8	Easy	1	0.99	0.87				1.00	1.00			
9	Hard	0		0.97		0.98	0.35		1.00		1.00	0.67
10	Easy	1	1.00	0.86				1.00	1.00			
11	Easy'	0	1.00		0.98			1.00		0.99		
12	Easy	1	1.00		0.90			1.00		0.98		
13	Hard	0		0.97		1.00	0.34		1.00		1.00	0.58
14	Easy	1	1.00	0.85				1.00	1.00			
15	Hard	1		0.97		0.97	0.34		1.00		1.00	0.51
16	Hard	1		0.99		0.99	0.79		1.00		1.00	0.88

Going through Table II in more detail, row 1 shows the KC estimates for SKT and CKT just before this sequence begins. Row 2 shows that an Easy problem was selected next. The estimates of only the KCs that are required for the composition step in that problem are shown. Even though the required KCs are above the 0.95 mastery threshold (at 0.98 and 0.997 respectively), the selection of an Easy problem is appropriate because there are other Area steps (not shown) in this problem (indicated as Easy', rather than just Easy) that are not above mastery (at 0.62). The student gets this composition step wrong (indicated by 0 in the Correct column). The updates for the relevant KCs can be seen in row 3 for both SKT (now 0.89 and 0.98) and CKT (now 0.94 and 0.99). Another Easy problem is selected (row 3), which is appropriate according to both models as the Compose KC is below .95 in both (.89 and .94). The student gets it right.

Now two easy problems are selected (rows 4 and 5) where area addition (AddAreas) is needed instead of area subtraction (SubtrAreas). The SKT estimate of AddAreas is below mastery for both problems, but goes above mastery before the second problem for the CKT estimate (see the bolded .97 vs. .91 in row 5). If problem selection had been driven by CKT, this problem would not have been selected and, arguably, the students' time would not have been wasted practicing mastered skills. (Note that the difference in the AddAreas estimates in Row 1 is caused by the difference in blame attribution on the one Hard problem the student saw before the data shown in Table II.) Rows 6-8 more clearly illustrate this difference in blame attribution. The student gets two consecutive Hard problems wrong and the SKT estimate of SubtrAreas drops to 0.87. However, it is likely that the student's difficulty is not with SubtrAreas, but with the SubGoal KC (knowing to find the areas of an irregular shape by finding the areas of the regular shapes

that make it up). Indeed, the CKT model puts most of the blame for these errors on SubGoal and little blame on SubtrAreas (which does drop slightly from .998 to .997).

4.2. Results: Fair blame assignment saves instructional time

To demonstrate that the example above is not idiosyncratic to the one student, we repeated the analysis illustrated above for all 120 students. We focused on the data from the first curriculum section where some steps are coded with multiple KCs (this is section 3 in Geometry Area unit). We used CKT to produce new KC estimates on each problem solved by each student as illustrated in Table II. We then identified the problems where all KCs involved were above the 0.95 mastery level according to the CKT estimates – like the 5 bolded problems in Table II. Of the 1370 problems, 441 or about 1/3 involved only mastered KCs according to CKT! If the problem selection had been driven by CKT, these problems would not have been given to students. These problems are likely to be unnecessary and are taking student time away from learning more difficult skills. (While the problem selection algorithm is designed to avoid giving mastered problems, 15 of the 1370 problems selected using SKT were mastered – still far below 441.)

Some of the 120 students, those with more prior knowledge, finished this section in as few as four problems (by getting all steps correct). Many others struggled and, like the student shown in Table II, got stuck in this thrashing between too many easier problems they tended to be able to solve and too few harder problems that exercised the composition (or subgoaling) skills they needed to acquire. The student in Table II is typical of these struggling students and, according to conjunctive knowledge tracing, five of the sixteen problems this student was given were unnecessary. For 33 of the struggling students, the tutor ran out of relevant problems and moved them on to the next section even though some KCs had not been mastered.

Current cognitive tutors have many steps coded with multiple KCs, for instance, in the algebra tutor some steps are coded with broad arithmetic skill categories (e.g., large vs. small numbers, rationals vs. whole numbers) in addition to the target algebraic skill. However, multiple KC coding occurs less often than it should. Doing so has often been avoided through the use of highly scaffolded interfaces, which have the downside of not assessing students in the unscaffolded context. Further, many steps that are currently coded with a single KC may be better modeled with multiple KCs [cf. Yudelso, Pavlik, and Koedinger, 2011].

5. DISCUSSION & CONCLUSIONS

We have presented an illustration of the problem of assigning blame when multiple knowledge components are required for an action and the student performs it incorrectly. A simple approach, currently used in practice, is to blame all components equally even though it may be just one (or some subset) that the student has not yet mastered. Until now, there has appeared to be little consequence to this simple approach. However, when we modified the problem selection algorithm to facilitate fading and unfading of problems with scaffolding, we found a negative consequence in the form of thrashing in problem selection. In the data from the Geometry Cognitive Tutor we found that real students were being assigned too many easy problems and not enough hard ones. Based on prior Bayesian student modeling work [Junker and Sijtsma 2001; Reye 1998; VanLehn et al. 1998], we adapted the standard knowledge tracing algorithm to create Conjunctive Knowledge Tracing (CKT), which provides a practical solution to fair blame assignment. CKT has the potential to make much better use of students' time in curricula that provide students with an adaptive learning trajectory from simple problems isolating key components of knowledge to difficult problems where multiple skills or concepts are required to produce a single response.

Alternative solutions to the blame assignment problem have been proposed [Conati, Gertner and VanLehn 2002; Pardos and Heffernan to appear; Reye 1998; VanLehn, Niu, Siler and Gertner 1998]. One simpler approach is to only blame the “hardest” KC, that is, the one with the lowest current probability. There are two potential limitations of this approach. First, if KCs are truly conjunctive and independent, such an approach will overly penalize the hardest KC and under penalize the others. We can see the difference in penalty in the KC values displayed in Row 1 of Table II (these values results from a failure on a hard problem just before this excerpt begins). Blaming only the hardest KC, which is SubGoal in this case, would yield a value of 0.49 (same as SKT would produce for this KC) whereas CKT yields a value of 0.70 (shown under Subgoal in the Conjunctive KT section). Thus, this blame-the-hardest approach could result in inappropriately requiring students to practice too many (harder) problems requiring the over-penalized KC and too few (easier) problems requiring the under-penalized KCs. A second limitation of the blame-the-hardest approach is that it does not facilitate the possibility of “unfading”, that is, of returning to scaffolded problems in the case that repeated failure on unscaffolded problems suggests (even with the softer penalty that CKT produces) the need to revisit easier problems.

Another simpler approach is to concatenate multiple KCs into a single combined KC. This approach has the downside that the student model has no information about knowledge overlap in related tasks and thus cannot be used in problem selection for the kind of gradual fading of scaffolding (going to harder problems when the student is ready) or reintroduction of scaffolding (going back to easy problems if needed) that is possible with CKT.

A more complex approach to the multiple-KC problem is to use a complete Bayesian network for the student model [e.g., Conati et al. 2002]. One immediate point of contrast with CKT is in the high effort required to engineer a student model as a Bayesian network. CKT can be relatively simply added to an existing model-tracing or constraint-based tutor as a plug-in, replacing the existing Knowledge Tracer if present. On the other hand, a full Bayesian network can represent dependencies between KCS and is not restricted to modeling KC learning only in terms of students direct experiences with those KCs. A Bayes net gives a modeler more freedom to hypothesize more complex interrelationships, like the learning of one KC enhancing another. Such freedom, however, may come at the loss of parsimony relative to the more constrained CKT approach whereby a set of KCs and a few direct computations on the KC parameter estimates may well represent all task difficulty and learning transfer relationships.

CKT is one solution within the broader space of Bayesian networks and Markov models for student modeling. As already mentioned, past work [Junker and Sijtsma 2001; Millán, Agosta and Pérez de la Cruz 2001] has articulated the multiplicative combination of noisy components. We have adapted this approach into the standard knowledge tracing by maintaining the Markov transition probability, but replacing the blame assignment with this multiplicative combination. Others have also incorporated the independence assumption and thus the multiplicative combination of components, but have put the noise (guess and/or slip parameters) at the level of the conjunction (sometimes called a “noisy-AND”) rather than at the level of the components [Conati et al. 2002]. In the psychometrics literature [Junker and Sijtsma 2001], the difference in whether the noise parameters are at the component level or the conjunction level is characterized by the contrast between the DINA (deterministic inputs noisy AND) and NIDA (noisy inputs deterministic AND) models. CKT is an extension of NIDA (adding the transition probability), with a slip and guess parameter for each conjunct in the AND. While the CKT and NIDA models have more parameters per AND relation than DINA, they can have fewer parameters in an overall student model in the case that there more AND

relations than components. For instance, there are four $(2n-n-1)$ possible AND relationships of three (n) components. Whether or not these theoretical differences make any practical difference will require future empirical comparison.

Whether and when CKT provides a more or less effective user model than more complex formulations such as Bayes nets will have to await future research. Nevertheless, an important contribution of this paper is the empirical evidence that comparing such alternatives is worth it. The problem selection thrashing we observed indicates that fair blame assignment can be a real problem and better solutions may have significant impact on student users of tutoring systems. The need for such a solution comes about in situations where we want a tutoring system to make dynamic and adaptive decisions about the fading of scaffolding or the “unfading” or reintroduction of scaffolding. Such capability would seem to be an important feature of a truly adaptive tutoring system and one that can be driven by educational data mining.

ACKNOWLEDGEMENTS

Thanks for support from the Pittsburgh Science of Learning Center (NSF-SBE #0354420; see learnlab.org), assistance from Carnegie Learning Inc. (carnegielearning.com) and the DataShop team, and support from the U.S. Department of Education (IES-NCSEER #R305B070487) and Ronald Zdrojkowski.

REFERENCES

- CONATI, C., GERTNER, A. AND VANLEHN, K. 2002. Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction* 12, 371-417.
- CORBETT, A.T. AND ANDERSON, J.R. 1992. Student modeling and mastery learning in a computer-based programming tutor. In *Intelligent Tutoring Systems: Second International Conference on Intelligent Tutoring Systems*, C. FRASSON, G. GAUTHIER AND G. MCCALLA Eds. Springer-Verlag, New York, 413-420.
- CORBETT, A.T. AND ANDERSON, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4, 253-278.
- GONG, Y., BECK, J. AND HEFFERNAN, N. 2010. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. In *Intelligent Tutoring Systems Conference Name*, V. ALEVEN, J. KAY AND J. MOSTOW Eds. Springer Berlin / Heidelberg, 35-44.
- HEFFERNAN, N. AND KOEDINGER, K. 1997. The composition effect in symbolizing: The role of symbol production vs. text comprehension. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, P. LANGLEY AND M.G. SHAFTO Eds. Lawrence Erlbaum Associates, 307-312.
- JUNKER, B.W. AND SIJTSMA, K. 2001. Nonparametric Item Response Theory in Action: An Overview of the Special Issue. *Applied Psychological Measurement* 25, 211-220.
- MILLÁN, E., AGOSTA, J.M. AND PÉREZ DE LA CRUZ, J.L. 2001. Bayesian student modeling and the problem of parameter specification. *British Journal of Educational Technology* 32, 171-181.
- PARDOS, Z.A. AND HEFFERNAN, N. to appear. Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *Journal of Machine Learning Research - Special Issue on the 2010 KDD Cup Competition*.
- PARDOS, Z.A., HEFFERNAN, N.T. AND RUIZ, C. 2008. Effective Skill Assessment Using Expectation Maximization in a Multi Network Temporal Bayesian Network. In *Proceedings of the The Young Researchers Track at the 9th International Conference on Intelligent Tutoring Systems*.
- REYE, J. 1998. Two-Phase Updating of Student Models Based on Dynamic Belief Networks. In *Intelligent Tutoring Systems*, B. GOETTL, H. HALFF, C. REDFIELD AND V. SHUTE Eds. Springer Berlin/ Heidelberg, 274-283.
- RITTER, S., ANDERSON, J.R., KOEDINGER, K.R. AND CORBETT, A. 2007. Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review* 14, 249-255.
- STAMPER, J. AND KOEDINGER, K.R. 2011. Human-machine student model discovery and improvement using DataShop. In *Proceedings of the 15th International Conference on AI in Education*.
- VANLEHN, K., NIU, Z., SILER, S. AND GERTNER, A. 1998. Student Modeling from Conventional Test Data: A Bayesian Approach without Priors. In *Intelligent Tutoring Systems Conference Name*, B. GOETTL, H. HALFF, C. REDFIELD AND V. SHUTE Eds. Springer Berlin / Heidelberg, 434-443.
- YUDELSON, M.V., PAVLIK JR., P. I. AND KOEDINGER, K.R. 2011. Towards Better Understanding of Transfer in Cognitive Models of Practice. In *Proceedings of the Fourth International Conference on Educational Data Mining*.