Technical Report: Worcester Polytechnic Institute <WPI-CS-TR-03-27>

A Methodology for Evaluating Predictions of Transfer and an Empirical Application to Data from a Web-Based Intelligent Tutoring System: How to Improve Knowledge Tracing in Dialog Based Tutors.

Neil T. Heffernan (nth@cs.wpi.edu)
Ethan A. Croteau (ecroteau@cs.wpi.edu)

Abstract:

Cognitive Science is interested in being able to develop methodologies for analyzing human learning and performance data. Intelligent tutoring systems need good cognitive models that can predict student performance. Cognitive models of human processing are also useful in tutoring because well-designed curriculums need to understand the common components of knowledge that students need to be able to employ (cite Koedinger paper and algebra stuff). A common concern is being able to predict when transfer should happen. We describe a methodology (first used by Koedinger, 2001) that uses empirical data and cognitively principled task analysis to evaluate the fit of cognitive models. This methodology seems particularly useful when you are trying to find evidence for "hidden" knowledge components that are hard to assess because they are confounded with accessing other knowledge components. We present this methodology as well as an illustration showing how we are trying to use this method to answer an important cognitive science issue.

Introduction

Koedinger and Junker's[1] insight invented the basic methodology that is described, extended and applied in this paper. Suppose you were a tutor that tried to get students ready to take the SAT (or some similar mathematics test.) Suppose your normal method was to present students a somewhat random SAT problem to see if they got it right. If they failed, you would provide some tutoring to make sure they eventually got the right answer. Let us suppose you wanted to know what other problems a student would do well on if they got practice on problems of a given type. Presumably, giving students practice on algebra problems will tend to transfer to other algebra problems (meaning that practice on algebra problems will make it more likely that that student will get other algebra problems correct), and might transfer, by a smaller amount, to geometry problems, but is unlikely to transfer to Verbal SAT Vocabulary problems due to the non-existent (presumably) overlap in the bits of knowledge between algebra problems and vocabulary problems. We desire a method that will allow us to build a model that predicts when transfer will happen between problems of a given type. We call this model a *Transfer model*. The better your transfer model, the more accurately you will be able to predict students' performance on different types of problems due to practice on certain other types of problems. If two problem types share no underlying knowledge then practice on either will make no difference in the average performance of the other problem type.

[1] The idea of evaluating a transfer model by looking for a parsimonious fit (using the Bayesian Information Criterion) to the data is due to Koedinger & Junker who shared this idea with me during my postdoc. They conceived of the idea of using two parameters in the logistic regression for each knowledge component. One of the parameters was used to indicate if the knowledge components were required. (We generalized the idea to not just the Boolean - present or absent - but to the number of times that knowledge component was used - zero, one, two, etc. times.) I am not familiar enough with the statistics literature to say these ideas are totally novel.

Related Work

Other researchers have attacked these problems, sometimes coming at the problems with a background in statistics. See Nichols, Chipman, & Brennan for a review of some of this work. In particular, work by Pirolli and colleagues (Draney, Pirolli, & Wilson, 1995, Pirolli, & Wilson, 1998) have addressed similar issues. Some work in Item Response Theory (IRT) (Hambleton & Swaminathan, 1985) is related (Embretson, & Reise, 2000). Multidimensional Item Response Theory bears more resemblance (Ackerman, 1996). Junker (1999) has analyzed the Draney, Pirolli & Wilson (1995) approach as well as an approach used by Corbett, Anderson & O'Brien (1995). The later is used by Corbett and Anderson (1999) who use the term *knowledge tracing* to indicate how their intelligent tutoring systems, which are used by thousands of students (see CarnegieLearing.com), track student's knowledge. The better job they do in correctly identifying the right knowledge components, the better they can give credit to students.

Definition of a Transfer Model

A transfer model is a very simple type of model and below we will discuss how it is different from other types of cognitive models (e.g., ACT-R models). A transfer model is a two dimensional array, in which problem types are listed on one side, and knowledge components are listed along the top. The elements in the array indicate whether a given knowledge component is required by a given problem type. If the component of knowledge is required, the number stored in the array indicates the number of times that knowledge component is required. Next we will give an example of a transfer model that we will use to illustrate our method.

Comparison of a Transfer Model and a Cognitive Model

Koedinger & MacLaren (2002) identify several constraints on cognitive models. Our *transfer model* can be used to address only two of the six constraints (Computational Parsimony, and Transfer). A *transfer model* does not address what Koedinger and MacLaren call Solution Sufficiency, Step Sufficiency, Choice Matching or Acquirability. These argue the need for a full-blown cognitive model (e.g., using ACT-R, Anderson, 1993) which is more costly to program and build. For one, a transfer model does not make a commitment about the order in which the knowledge components are linked together (which can be a benefit). Also, the transfer model requires no commitment with regard to thinking of the knowledge components as declarative or procedural (i.e., rules). One of the limitations is that we are not modeling how two components of knowledge could be competing with each other (in ACT-R you could model this). Nor are we modeling how context plays a role; for instance, when applying the *articulating variable* knowledge component it might be harder in some cases than others. You can do similar things with cognitive models (see Baker, Corbett, & Koedinger, 2003). Yet another limitation is that we are not modeling forgetting, while it is clear how to do this in ACT-R, which has explicit support.[2]

Yet another simplification our model makes is in our prediction of maximally complete transfer of the individual knowledge component; yet we know that if the knowledge components are not fine-grained enough, we should not expect to see the transfer happen in this maximally predicted manner. Transfer is notoriously difficult to get to happen, as well as to predict when it occurs.

---

[2] We ignore, particularly, the type of forgetting and relearning that happens over days, such as the sort of thing that happens when students log out of the tutor for the weekend; on Monday they will have initially forgotten some knowledge components but can relearn what learned previously but in a shorter period of time.

In summary, we see a use for the simpler transfer model due to its simplicity, while recognizing some of its disadvantages. It might be that using a transfer model is a good first step to building a cognitive model.

Example of a Transfer Model

We were interested in investigating a particular domain that one of the authors studied (Heffernan, 2001, Koedinger & Heffernan, 1997, 1998, Heffernan & Koedinger, 2002) and has continued to report learning results (Heffernan, 2003). Heffernan hypothesized the existence of a hidden knowledge component that will be explained below. We wanted to see if we could find evidence to support the existence of this hidden knowledge component by applying a transfer model to a set of tutorial log files (the collection of which is reported in Heffernan, 2003). This process of applying a transfer model to performance data (captured in the form of tutorial log files) will be explained after we first give an extended example of a transfer model. This transfer model, presented in Table 1 was developed to predict difficulty and transfer on different versions of problems that required students to *symbolize* (i.e., read an algebra *word problem* and then write a mathematical expression). The justification and explanation of the numbers in Table 1 will be explained below.

		Example Problem Type and Answer	Arithmetic	Comprehending One-Step	Compose Concrete	Articulating One-Step	Articulating Variable	Articulating Composed Expression
			Knowledge Components					
Question Types	1 Step Compute	Given a word problem that represents 5+3; answer= 8.	1	1	0	0	0	0
	2 Step Compute	Given a word problem that represents (10-4)/2; answer= 3.	2	2	1	0	0	0
	1 Step Articulation	Given a word problem that represents 5-3; answer= 5-3.	0	1	0	1	0	0
	2 Step Articulation	Given a word problem that represents 4/7+6; answer= 4/7+6.	0	2	0	2	0	1
	1 Step Symbolization	Given a word problem that represents 5-x; answer= 5-x.	0	1	0	1	1	0
	2 Step Symbolization	Given a word problem that represents 4/z+6; answer= 4/z+6.	0	2	0	2	1	1

Table 1: A Transfer Model that relates the knowledge components needed on six different problem types. A number in a cell indicates the number of times that column's knowledge component is used in that row's problem type.

The Question Types

Heffernan defined six different question types. The six question types were the result of crossing two different *factors*. The first factor we call the *task directions* factor. Here are examples showing the three different versions of the factors we call *task directions*.

1. **Compute:** Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for <u>3 minutes</u> and stops to rest. How far is she from the dock now?
2. **Symbolize:** Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for <u>"m" minutes</u> and stops to rest. How far is she from the dock now?
3. **Articulate**: Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for <u>3 minutes</u> and stops to rest. <u>Can you write an expression that will compute</u> how far is she from the dock now?

Observe that the only difference (indicated with underlining) between the compute question and the symbolize task is that there is the variable, "m", in place of the constant "3" minutes. The only difference between the compute and the articulate task was that students needed to write the complete expression, and not just calculate the answer. We checked that students understood the directions before hand (Heffernan & Koedinger, 1997).

Heffernan & Koedinger also defined a factor, which we will call *steps,* that simply indicates the number of math operators needed to solve the problem (See examples in Table 1). Here are examples showing two different versions of the factors we call *steps,* again with underlining added to highlight the difference.

1. **One Step:** Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far <u>did she row</u>?
2. **Two Step:** Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far <u>is she from the dock now</u>?

Next we look at the knowledge components that were hypothesized to be at work that could explain the performance data that was observed.

The Knowledge Components

The six knowledge components that Heffernan & Koedinger identified were:

1) **Arithmetic –** This component requires doing any one mathematical operation.
2) **Comprehending One-Step –** This component requires extracting the operator and two numbers from the word problem. It represents the parsing of a part of the word problem. It represents everything you need to do for a one-step arithmetic word problem excluding the actually computation. This component applies in all the problem types.
3) **Composing Concrete**: While doing a two-step arithmetic problem, a student needs to learn to remember a computed value and later use that value for the 2^{nd} step. This is meant to deal with the fact that computing a two-step problem is harder than two one-step arithmetic problems.
4) **Articulating One-Step**: The ability to write a math expression that has already been assembled as a mental representation.
5) **Articulating Variable**: Appropriately handling a variable in the problem.
6) **Articulating Composed Expression**: This component is the component that allows a student to treat an articulated expression the same way a number is treated.

The sixth component is the component that we were particularly interested to try to determine if it really exists, as hypothesized by Heffernan & Koedinger (1998). This is the hidden component that we will test to see if its incorporation creates a better model.[3]

Motivation for This Particular Domain

Many researchers had argued that the students have difficulty with symbolization because they have trouble comprehending the word in an algebra word problem. For instance, Nathan, Kintsch, & Young (1992) "claim that [the] symbolization [process] is a highly *reading-oriented* one in which poor *comprehension* and an inability to access relevant long term knowledge leads to serious errors. [emphasis added]". However, Heffernan & Koedinger showed that many students can do *compute* tasks well, whereas they have greater difficulty with the *symbolization* tasks. This showed that many could comprehend the words in the problem, yet still could not do the symbolization. Other researchers argued that the hard part of doing symbolizations was the presence of the variable, so symbolization tasks should be the hardest. However, Heffernan & Koedinger (1997, 1998) presented evidence that showed that there is hardly any difference between students' performance on *articulate* tasks and *symbolization* tasks, debunking the idea

[3] In point of fact, we could apply the same test to any of the knowledge components by removing the column, or changing the numbers in the table based upon some cognitively plausible theory. Koedinger, Junker and Heffernan have already proposed and implemented a search method, but that will not be explained in this paper.

that the hard part is the presence of the variable per se. Instead, Heffernan & Koedinger found the main difficulty was between *compute* tasks and *articulate* tasks, which they explained by saying that the main knowledge component that students were missing was the ability to articulate the mathematics. Students might know what to do, but fail to symbolize correctly, simply because they don't know how to articulate the steps in the foreign language of algebra. There seemed to be a slight interaction in that students doing two-step articulation problems were unusually hard, suggesting that articulating out the two steps at the same time was particularly different. Heffernan & Koedinger argued that this was maybe due to students not knowing the algebra grammar rule that says an expression is an

<expression> = <expression> <operator> <expression>

as opposed to

<expression> = <number> <operator> <number>

See Heffernan 2001 for a more intensive discussion of this interpretation.

Understanding This Particular Transfer Model

We will briefly review the knowledge components and why the particular numbers in Table 1 make for a cognitively plausible interpretation.

To understand what the model in Table 1 hypothesizes, let us look at the first row. The first row says that a student presented with a one-step compute question will have to use an *arithmetic* knowledge component once and a *comprehending one-step* knowledge component once.[4] Heffernan & Koedinger (1997, 1998) showed that doing two one-step compute tasks is harder than doing one two-step compute task, so we knew that the transfer model would need a new component of knowledge to explain the additional difficulty. We chose to model that difficulty by adding a new knowledge component that we call *composing concrete*.

The second row in Table 1 predicts that a student would have to (here we give a logical ordering to the sequence of knowledge components that is not implied by the model) read the problem and extract the operations twice (using *comprehending one-step* twice) followed by doing the math for the first operation (*arithmetic* once) followed by remembering the value computed in the first step (represented by the *composing concrete* knowledge component) and finally using that value in the last step (the second usage of the *arithmetic* knowledge component).

Since Heffernan & Koedinger found that articulating a single step is harder than computing a single step, we added the knowledge component we call *articulating one-step*.

Because we found there is little difference between the knowledge required to perform an articulating one-step and symbolizing one-step problem (the difference was not statically significant), we could have chosen to leave out the *articulating variable* knowledge component. However we decided to leave this knowledge component in our transfer models.[5]

Because we hypothesized that articulating a two-step expression might be more difficult than an expression with two one-step articulations we added a 6th knowledge component, which we called *articulating composed expression*. In the previous section we explored a few different interpretations for this knowledge component.

Understanding How This Model Predicts Transfer

Qualitatively, we can see that the transfer model in Table 1 predicts that practice on one-step compute questions should transfer to one-step articulation problems only to the degree that a

---

4 When it is clear from context, we will stop saying "knowledge component" and just refer to the name of the knowledge component in italics.

5 As future work we will explore if it really belongs.

student learns (i.e., get practice at employing) the *comprehending one-step* knowledge component. We can turn this qualitative observation into a quantified prediction method by treating each knowledge component as having a *difficulty* factor and a *learning* factor.

One of the most ubiquitous findings in learning research is known as the Power Law of Learning (Anderson, 1993). This law says that speed (and chance of recalling a knowledge component) increases as a power function of the number of practice opportunities. The basic equation for a power law is $y = a + bx^{-d}$, where "*a*" is the asymptote (minimal time to perform a step), "*b*" is a scaling constant, and "*d*" is the learning rate. We will ignore the scaling factor and work with just two parameters.[6] The "a" corresponds with our *difficulty* factor, while "d" corresponds to our *learning* factor. We will assume that the Power Law of Learning applies to each of these knowledge components separately, because this law says that we should be able to fit our learning data to a power function. Intuitively, this means that you should be able to see a smooth progression of learning. This can be shown by plotting the response time for a question that involves only one knowledge component, versus the number of previous attempts at that knowledge component.

Junker, Koedinger, & Trottini (2000) showed that a logistic regression was the right way to incorporate the Power Law of Learning through using statistics. Intuitively, we are trying to use our data to get smooth learning curves. Each knowledge component will get two parameters in the logistic regression. One of the parameters we call the *difficulty* parameter and the other we call the *learning* parameter. The *difficulty* parameter tells the number of times the knowledge component is used for a given row in the data set,[7] while the *learning* parameter keeps track of the number of times the student has previously encountered that knowledge component. In essence, the *difficulty* parameters indicate the incoming knowledge students possess of the knowledge components. *Learning* parameters that have a high coefficient in the logistic regression are learned quicker than those learning parameters having a lower coefficient. The *learning* parameter indicates the steepness of the learning curve. All *learning* parameters should ideally be positive, indicating that as students practice they are more likely to get a correct response.[8]

Using the Transfer Model to Predict Transfer in Tutorial Log Files

Heffernan (2001) created Ms. Lindquist, an intelligent tutoring system, and put it on line (***www.algebratutor.org***) and collected tutorial log files for all the students learning to symbolize. Table 2 shows an example of such a dialog, and in the next paragraphs will explain all of the complexity of Table 2. Ms. Lindquist is a system that provides coached-practice in that it presents problem *scenarios* to students and then asks the student to symbolize an expression. We will use the term *scenario* to refer to the individual word problems, and the term *question* to refer to the individual questions that get asked at each step in the dialog. Table 2 shows mostly a student working on the Scenario #1 (The scenario identification number is listed in the first column) beginning "Anne is rowing a boat…". The last few rows of Table 2 show a new scenario, Scenario #2, which begins with "Michael works as a waiter". Each word-problem has a single *top-level* question. This top-level question is always a *symbolize* question. If the student

---

[6] Future research could attempt to fit each knowledge component with three instead of four components, but right now we are only using two. We could also apply the power law of forgetting to modeling forgettin g which also occurs more with time fitting a power function.

[7] Each row in the data set looks like a row in Table 2. The dataset used for the logistic regression combines a single question provided by the tutor with the accompanying student's response. F or instance, the first two rows from Table 2 provide a single row of data for our logistic regression. In the logistic regression, the response time and correctness are the dependent variables, whereas the difficulty and the learning parameters are the independent variables.

[8] Currently we do not force the logistic regression to insure this.

fails to get the top level question correct, Ms. Lindquist steps in to have a dialog (as shown in the 6th column) with the student, asking questions to help break down the problem into simpler questions.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
		Question Types					Outcomes		Knowledge Components: *difficulty* parameters						Knowledge Components: *learning* parameters					
	Scenario Identifier	**Factor:** *Task directions*	**Factor:** *Steps*	**Attempt at problem type**	**Hint Level**	**Example Question**	**Response Time (Seconds)**	**Response (Correct or not)**	**Arithmetic**	**Comprehending One-Step**	**Compose Concrete**	**Articulating One-Step**	**Articulating Variable**	**Articulating Composed Expression**	**Arithmetic**	**Comprehending One-Step**	**Compose Concrete**	**Articulating One-Step**	**Articulating Variable**	**Articulating Composed Expression**
1	1	S	2	1	0	**Scenario 1:** Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far is she from the dock now? **Tutor**: Please write an expression for the distance Anne has left to row.			**0**	**2**	**0**	**2**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
2						**Student**:8-40/m	**130**	**N**												
3	1	C	2	1	0	**Tutor**: Hmm. No. We will come back to the distance Anne has left to row. Let me try to break this down for you. Please compute the distance Anne has left to row if the number of minutes rowing was 3 minutes?			**2**	**2**	**1**	**0**	**0**	**0**	**0**	**2**	**0**	**2**	**1**	**1**
4						**Student**:580	**38**	**N**												
5	1	C	1	1	0	**Tutor**: Hmm. No. Let me ask you a different question. Please compute the distance rowed so far if the number of minutes rowing was 3 minutes?			**1**	**1**	**0**	**0**	**0**	**0**	**2**	**4**	**1**	**2**	**1**	**1**
6						**Student**:12	**19**	**N**												
7	1	C	1	2	1	**Tutor**: Hmm. No. Try again please. Please compute the distance rowed so far if the number of minutes rowing was 3 minutes? Hint: Use the fact that her speed rowing is 40 yards per minute and the number of minutes rowing is 3 minutes.			**1**	**1**	**0**	**0**	**0**	**0**						
8						**Student**: 120	**9**	**Y**												
9	1	C	2	2	0	**Tutor**: Correct. Now, lets try this question again: Please compute the distance Anne has left to row if the number of minutes rowing was 3 minutes?			**2**	**2**	**1**	**0**	**0**	**0**						
10						**Student**: 680	**24**	**Y**												
11	1	A	2	1	0	**Tutor**: Correct. Using only 800, 40, and 3 write an expression for the distance Anne has left to row			**0**	**2**	**0**	**2**	**0**	**1**	**3**	**5**	**1**	**2**	**1**	**1**
12						**Student:** 120-800	**18**	**N**												
13	1	A	1	1	1	**Tutor:** No. Your answer of "120-800" equals "-680", but you wanted to write an expression that equaled "680". But that is only because you didn't say where the 120 came from. What math did you do to get the 120?			**0**	**1**	**0**	**1**	**0**	**0**	**3**	**7**	**1**	**4**	**1**	**2**
14						**Student:** 3*40	**22**	**Y**												
15	1	A	2	2	0	**Tutor:** Correct. Now, lets try this question again: Using only 800, 40, and 3 write an expression for the distance Anne has left to row.			**0**	**2**	**0**	**2**	**0**	**1**						
16						**Student**: 800-40*3	**18**	**Y**												
17	1	S	2	2	0	**Tutor**: Correct. Now use "m" instead of 3 to again express the distance Anne has left to row.			**0**	**2**	**0**	**2**	**1**	**1**						
18						**Student**:800-40*m	**9**	**Y**												
19						**Tutor**: Correct. Lets do a new problem.														
20	2	S	2	1	0	**Scenario 2:** Michael works as a waiter. Saturday he worked "h" hours. The restaurant pays him an hourly wage of $3 an hour. He also made $40 dollars in tips. Write an expression for the total amount he earned Saturday Night. **Tutor:** Write an expression for the total amount he earned Saturday Night.			**0**	**2**	**0**	**2**	**1**	**1**	**3**	**8**	**1**	**5**	**1**	**2**

Table 2: Showing a made-up tutor log file and how it uses the Transfer Model.

The second and third columns show the question types. The second column uses for *Task direction* either S=Symbolize, C=Compute or A=Articulate. By crossing *task direction* and *steps* there are six different question types, as previously listed in Table 1. The fourth column defines what we call the attempt. The first time, for each scenario, one of the six question types is asked, the attempt for that question type is set to one. Notice that in line 20, that attempt is reset for a two-step symbolization question because a new scenario is being used. The 5th column shows the hint level, which is incremented if the student is given a question that is simply a rephrasing of a question they were already asked. (In this sense, we say that a new question is not being asked, unless the answer that the student is supposed to type is different.) The 6th column has the exact dialog that appeared in the log file. The 7th and 8th columns are grouped together because they are both outcomes that we will try to predict.[9] Columns 9-14 show the *difficulty* parameters for each knowledge component, while columns 15-20 show the *learning* parameters. The *difficulty* parameter is taken straight from the row from Table 2, which matches the question type (Columns 2 crossed by column 3). The *learning* parameter is calculated on the fly, and counts the previous attempts that have been made at that knowledge component by that person. The learning parameter keeps track of the number of previous attempts to assess a component. (It could be called the count of previous attempts) Notice that the learning knowledge component is **not** reset on line 20 when a new scenario was instructed, but notice the attempts **is** reset. Notice that on lines 7, 9, 15 and 17, there are no learning values, and that is because we have decided to exclude from our analysis any question that appears after the first time for each scenario.

How a Transfer Model was Used to Predict Transfer

We had access to hundreds of instances of students learning a particular component which we then were able to use to look for the predicted transfer. Specifically, we started with a dataset from Mr. X's students, whose learning results were previously reported in Heffernan (2003).

Experiment # 1

In this experiment we want to determine if our data shows evidence of the hidden skill, which we mentioned above as being the *articulating composed expression* component.

Mr. X had 76 students use Ms. Lindquist. Each student's tutoring session was logged to a flat text file, which we use to facilitate our study. Constructing a parser for these student files was necessary to get the data in a usable format. The data generated for our study was in a tab delimited file. There were 34 fields selected, which would represent a single row of data. Among these fields were the student's name, problem name, correct answer, question type, etc.

The data produced from the initial parsing contained data that could not directly be used for the analysis. This was remedied by cleaning the data using a filter to remove unnecessary rows. Students who selected to do problems from the "demo" section had to have these specific problems removed, because they were only for demonstrative purposes of the teaching strategies employed by the tutor. It was determined that only problems from the first two tutorial sections would be used, since many students had not completed problems in the third and forth sections. Problems that employed the "verbal" strategy were determined to not be useful for the analysis. It was also determined that only the first attempt for a given problem would be included. A first attempt is defined as the first occurrence of a problem or sub problem having a correct answer that has not yet been asked. This decision was made, because successive attempts would not be

[9] Currently, we are only predicting whether the response was correct or not, but later we will do a Multivariate logistic regression to take into account the time required for the student to respond.

useful for showing the learning taking place using our current methodology. Finally it was determined that a student should have answered a minimum of four problems in the second section to have used the tutor for a sufficient amount of time.

The filtered dataset consisted of 1460 rows of data, which encompassed 73 students. This means that only 3 students had been excluded from the dataset, because they had not completed enough problems in the second section. Each row of data represented a student's response for which we wanted to find the best fitting model. To do this we planned to run a logistic regression to predict the probability that a student would get the correct answer on any given first attempt. The dependent variable of the logistic regression was Boolean to indicate if the student's response was correct. The independent variables of the logistic regression were the students as well as two parameters for each component in the model. The first parameter was Boolean and indicated if the component was present for that particular problem, which was determined by our cognitive model. The second parameter indicated the number of times that particular component had been seen by that student up until that point. Intuitively, the two parameters determined the shape of the learning curve. The first parameter determined how difficult the component was for a student, whereas the second parameter indicated how steep a learning curve there was for that component.
The dataset[10] is available at http://www.cs.wpi.edu/~ecroteau/data/mrx/

		Steps		
Qtype	Data	1	2	Grand Total
QCOMPUTE	Count of Number Done	80	91	171
	Total Correct	26	7	33
	Average % Correct	0.325	0.076923077	0.192982456
QEXPLAIN	Count of Number Done	34	74	108
	Total Correct	33	44	77
	Average % Correct	0.970588235	0.594594595	0.712962963
QSYMB	Count of Number Done	435	746	1181
	Total Correct	341	379	720
	Average % Correct	0.783908046	0.508042895	0.609652837
Total Count of Number Done		549	911	1460
Total Correct		400	430	830
Total Average % Correct		0.72859745	0.472008782	0.568493151

Results

The logistic regressions were produced using S-PLUS 6.1. Two models were created, one without the proposed hidden component and the other with the proposed hidden component. The model with the hidden component had two additional independent variables, one indicating the presence of the hidden component and the second indicating the number of times the component had previously been presented. See Appendix A for the coefficients and Bayesian Information Criterion (BIC) obtained for each logistic regression, which should be used as reference in the following discussion. The measure chosen to compare how well each model fit our data was its BIC value.

[10] Two datasets are provided. The dataset containing only first attempts was used for our logistic regression. The dataset with all attempts is also provided, because it provides a better understanding of the student/tutor interaction.

Discussion

The BIC of the second model (the model with the hypothesized hidden skill) was greater by about six (less is better). This suggests that our transfer model was not improved by including the *articulating composed expression* component. The difference of fitness in the transfer models was not statistically significant since the difference was less than ten.

Taking a look at the coefficients from both models, the arithmetic component has the lowest coefficient. This can be interpreted as the arithmetic component being the hardest to learn since it decreased the probability of a student getting the correct response more than the other components. Looking at the coefficients relating to a student's gained experience from practice with the tutor (component order), both models indicate that the arithmetic component improved far quicker than the other components through practice. The existence of negative coefficients for some of the component orders is surprising, as this would suggest that a student did worse on these particular components as their practice increased. It is possible that the logistic regression over fit the data by determining the best possible fit without taking into account the constraints imposed by our model, that a student should typically become better a using a component through practice with using that component.

Conclusion

Although the analysis of the students' interaction with the tutor does not lend evidence to the existence of the hidden component, it is not statistically significant that the transfer model is better without the hidden component. Possibly restricting the logistic regression to having only positive coefficients for the component orders would have suggested otherwise.

Future Work:

An important aspect of the student's learning was overlooked by our transfer models. This oversight was the learning that takes place from receiving feedback on a question, which is in turn related to the tutor's following questions. This can be illustrated by looking at two adjacent questions in Table 2. An example will now be provided to clarify this concept. When assigning the learning parameters in the above Table 2, it is first necessary to have the difficulty parameters associated with the question type. In our current scheme, we have a one-to-one mapping of question type to difficulty parameters. Using a new scheme, the difficulty parameters would be determined based on what the student has previously demonstrated for knowledge components on the same problem. By doing this, a student would not receive credit for additional learning (the learning parameters) for demonstrating a knowledge component that had previously been exercised on another question which was part of the same problem. Now to clarify this concept, we will look at Table 3, which indicates the first two questions presented by the tutor for this particular scenario.

		Question Types					Outcomes		Knowledge Components: *difficulty* parameters						Knowledge Components: *learning* parameters					
	Scenario Identifier	**Factor:** *Task directions*	**Factor:** *Steps*	**Attempt at problem type**	**Hint Level**	**Example Question**	**Response Time (Seconds)**	**Response (Correct or not)**	**Arithmetic**	**Comprehending One-Step**	**Compose Concrete**	**Articulating One-Step**	**Articulating Variable**	**Articulating Composed Expression**	**Arithmetic**	**Comprehending One-Step**	**Compose Concrete**	**Articulating One-Step**	**Articulating Variable**	**Articulating Composed Expression**
1	1	S	2	1	0	**Scenario 1:** Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far is she from the dock now? **Tutor**: Please write an expression for the distance Anne has left to row.			**0**	**2**	**0**	**2**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
2						**Student**:8-40/m	**130**	**N**												
3	1	C	2	1	0	**Tutor**: Hmm. No. We will come back to the distance Anne has left to row. Let me try to break this down for you. Please compute the distance Anne has left to row if the number of minutes rowing was 3 minutes?			**2**	**~~2~~ 0**	**1**	**0**	**0**	**0**	**0**	**2**	**0**	**2**	**1**	**1**
4						**Student**:580	**38**	**N**												

Table 3: Illustrating proposed modification to determine the updating difficulty parameters

In the first question, the difficulty parameters would remain as previously stated, because no knowledge components have yet been demonstrated for this problem (the learning parameters are all zero). For the second question, the difficulty parameters would be slightly modified using this new scheme. The *Comprehending One-Step* difficulty parameter would be zero instead of two since there is overlap in these two knowledge components, which are identical. The reasoning behind this introduced complication is that the difficulty parameters are to represent a question's difficulty. In doing this, they must represent a student's knowledge of the current question, which is largely dependent the feedback received from the tutor when doing previous questions as part of the same problem.

Other work includes adding additional dependent variables to the logistic regression, such as time to answer. In such a multivariate regression an optimization of student's response and time would be made. This would be fairly interesting to see and the data required is already available in the dataset used for this experiment. Another possibility is to examine how the hints provided by the tutor effect the transfer. This would allow generalizations to be made on the relative effectiveness of the various tutorial strategies made available by the tutor.

References

Ackerman, T. A. (1996). Graphical representation of multidimensional item response theory analyses. Applied Psychological Measurement, 20, (4) 311-329.

Anderson, J. R. (1993). *Rules of the Min*d. Hillsdale, NJ: Erlbaum.

Baker, R.S., Corbett, A.T., Koedinger, K.R. (2003) Statistical Techniques for Comparing ACT-R Models of Cognitive Performance. Presented at 10th Annual ACT-R Workshop.

Corbett, A. T., Anderson, J. R., & O'Brien, A. T. (1995) Student modeling in the ACT programming tutor. Chapter 2 in P. Nichols, S. Chipman, & R. Brennan, *Cognitively diagnostic assessment*. Hillsdale, NJ: Erlbaum.

Draney, K. L., Pirolli, P., & Wilson, M. (1995). A measurement model for a complex cognitive skill. In P. Nichols, S. Chipman, & R. Brennan, *Cognitively diagnostic assessment*. Hillsdale, NJ: Erlbaum.

Embretson, S. E. & Reise, S. P. (2000) Item Response Theory for Psychologists Lawrence Erlbaum Assoc.

Hambleton, R. K., & Swaminathan, H. (1985). Principles and applications of item response theory. Boston, MA: Kluwer-Nijhoff Publishing Company.

Heffernan, N. T. (2001). *Intelligent Tutoring Systems have Forgotten the Tutor: Adding a Cognitive Model of an Experienced Human Tutor*. Dissertation & Technical Report. Carnegie Mellon University, Computer Science, http://www.algebratutor.org/pubs.html.

Heffernan, N. T. (2003) *Web-Based Evaluations Showing both Cognitive and Motivational Benefits of the Ms. Lindquist Tutor* 11th International Conference Artificial Intelligence in Education. Syndey. Australia.

Heffernan, N. T., & Koedinger, K. R.(1997) The composition effect in symbolizing: the role of symbol production versus text comprehension. In *Proceeding of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 307-312). Hillsdale, NJ: Lawrence Erlbaum Associates.

Heffernan, N. T., & Koedinger, K. R. (1998) A developmental model for algebra symbolization: The results of a difficulty factors assessment. *Proceedings of the Twentieth Annual Conference of the Cognitive Science Societ*y, (pp. 484-489) Hillsdale, NJ: Lawrence Erlbaum Associates.

Heffernan, N. T., & Koedinger, K. R.(2002) *An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor* International Conference on Intelligent Tutoring System 2002. Biarritz, France.

Junker, B. (1999) Some statistical models and computational methods that may be useful for cognitively-relevant assessment. Prepared for the Committee on the Foundations of Assessment, National Research Council, November 30, 1999. http://www.stat.cmu.edu/~brian/nrc/cfa/documents/final.pdf

Junker, B., Koedinger, K. R., & Trottini, M. (2000). Finding improvements in student models for intelligent tutoring systems via variable selection for a linear logistic test model. Presented at the Annual North American Meeting of the Psychometric Society, Vancouver, BC, Canada. http://lib.stat.cmu.edu/~brian/bjtrs.html

Koedinger, K.R., & MacLaren, B. A. (2002). Developing a pedagogical domain theory of early algebra problem solving. CMU-HCII Tech Report 02-100. Accessible via http://reports-archive.adm.cs.cmu.edu/hcii.html.

Nathan, M. J., Kintsch, W. & Young, E. (1992). A theory of algebra-word-problem comprehension and its implications for the design of learning environments. *Cognition & Instruction* 9(4): 329-389.

Pirolli. P., & Wilson, M. (1998). A theory of the measurement of knowledge content, access, and learning. *Psychological Review* 105(1), 58-82.

Raftery, A.E. (1995) Bayesian Model Selection. Sociological Methodology, 111-196.

Appendix A

MODEL WITHOUT HIDDEN SKILL

Call: glm(formula = tutor$response ~ tutor$student + tutor$arithmatic + tutor$comprehend.one.step + tutor$composing.concrete.number.in.head + tutor$articulating.one.step + tutor$articulating.variable + tutor$sko.arithmatic + tutor$sko.comprehend.one.step + tutor$sko.composing.concrete.number.in.head + tutor$sko.articulating.one.step + tutor$sko.articulating.variable, family = binomial)

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.481427	-0.9140756	0.4355614	0.8674109	2.820983

Coefficients: (2 not defined because of singularities)

	Value	Std. Error	t value
(Intercept)	4.4988125865	0.45048997	9.98648772
tutor$student1	-0.7113014438	0.35831152	-1.98514812
tutor$student2	0.2352825547	0.20285129	1.15987702
tutor$student3	-0.4697257122	0.19392836	-2.42216105
tutor$student4	0.4356926723	0.14051066	3.10078022
tutor$student5	0.0194086202	0.12209825	0.15895904
tutor$student6	0.0559032683	0.10067235	0.55529913
tutor$student7	-0.0750249298	0.14265622	-0.52591418
tutor$student8	0.0953674549	0.07056179	1.35154537
tutor$student9	-0.0531032628	0.04735749	-1.12132773
tutor$student10	-0.0755156141	0.12308641	-0.61351706
tutor$student11	0.1355240872	0.05657884	2.39531402
tutor$student12	0.0857907693	0.04597711	1.86594509
tutor$student13	0.1489550559	0.05719988	2.60411499
tutor$student14	0.0273215075	0.03576432	0.76393196
tutor$student15	0.0182932183	0.03833542	0.47718835
tutor$student16	0.0613498850	0.03405556	1.80146438
tutor$student17	0.0533744924	0.03254365	1.64008906
tutor$student18	-0.0006749371	0.02522333	-0.02675845
tutor$student19	-0.0281935125	0.02528251	-1.11513913
tutor$student20	0.0008747127	0.02195746	0.03983669
tutor$student21	-0.0326691402	0.05246953	-0.62263075
tutor$student22	0.0951625740	0.03073715	3.09601178
tutor$student23	-0.0021173286	0.02079953	-0.10179692
tutor$student24	0.0310542336	0.02276412	1.36417471
tutor$student25	0.0413856451	0.02412238	1.71565339
tutor$student26	-0.0319236584	0.02412207	-1.32342124
tutor$student27	0.0166471559	0.02230054	0.74649131
tutor$student28	0.0374648064	0.02181816	1.71713874
tutor$student29	0.0399272471	0.019704411	2.02631012
tutor$student30	0.0100286957	0.016093472	0.62315303
tutor$student31	0.0394725886	0.020810004	1.89680829
tutor$student32	0.0404408891	0.022750847	1.77755535
tutor$student33	0.0236924364	0.016323691	1.45141417
tutor$student34	0.0405185734	0.018365362	2.20624968
tutor$student35	0.0170595016	0.016656424	1.02419953
tutor$student36	-0.0064228320	0.013625948	-0.47136771
tutor$student37	0.0001040380	0.013566451	0.00766877
tutor$student38	0.0011880672	0.022226865	0.05345186
tutor$student39	0.0358792609	0.017943288	1.99959237
tutor$student40	0.0048532279	0.012760793	0.38032337
tutor$student41	-0.0245880362	0.011855959	-2.07389683
tutor$student42	0.0140686754	0.014343635	0.98083052
tutor$student43	-0.0079348055	0.014795684	-0.53629189
tutor$student44	-0.0089381779	0.010361984	-0.86259326
tutor$student45	0.0320913002	0.014983637	2.14175635
tutor$student46	0.0231322100	0.013018651	1.77685151
tutor$student47	-0.0003008711	0.014010907	-0.02147407
tutor$student48	0.0142292259	0.020237540	0.70311045
tutor$student49	0.0316170238	0.016910750	1.86964050
tutor$student50	-0.0219988246	0.012144663	-1.81139846
tutor$student51	-0.0029708805	0.009296063	-0.31958480
tutor$student52	-0.0119906307	0.009280664	-1.29200140
tutor$student53	0.0223996697	0.010431516	2.14730725
tutor$student54	0.0013780870	0.011376987	0.12112935
tutor$student55	0.0059958869	0.011182893	0.53616598
tutor$student56	-0.0119658967	0.008219435	-1.45580523
tutor$student57	-0.0273008464	0.009093140	-3.00235631
tutor$student58	0.11233805498	0.072149137	1.55702563
tutor$student59	-0.01517286269	0.010654711	-1.42405203
tutor$student60	-0.03066491498	0.006976070	-4.39572945
tutor$student61	-0.00070700272	0.010734132	-0.06586492
tutor$student62	0.00672114550	0.011768628	0.57110696
tutor$student63	0.01869471292	0.011323023	1.65103545
tutor$student64	-0.01594613455	0.007921654	-2.01298033
tutor$student65	-0.01436328701	0.006714989	-2.13898905
tutor$student66	-0.02546359005	0.007049230	-3.61225143
tutor$student67	0.01449139255	0.010824889	1.33871055
tutor$student68	0.00072458547	0.007864355	0.09213540
tutor$student69	-0.02907487973	0.009681241	-3.00321839
tutor$student70	0.00009422061	0.008835518	0.01066385
tutor$student71	0.00909919057	0.008680405	1.04824492
tutor$student72	-0.00268564287	0.007181876	-0.37394725
tutor$arithmatic	-3.51648116812	0.436577089	-8.05466264
tutor$comprehend.one.step	-2.32601735179	0.250476556	-9.28636750
tutor$composing.concrete.number.in.head	3.76950883996	0.885386360	4.25747336
tutor$articulating.one.step	NA	NA	NA
tutor$articulating.variable	-0.65625018185	0.277606809	-2.36395564
tutor$sko.arithmatic	-0.10663701997	0.216595221	-0.49233321
tutor$sko.comprehend.one.step	0.02154295520	0.034058690	0.63252449
tutor$sko.composing.concrete.number.in.head	0.30730119243	0.597691656	0.51414670
tutor$sko.articulating.one.step	NA	NA	NA
tutor$sko.articulating.variable	0.03432116642	0.061790679	0.55544245

(Dispersion Parameter for Binomial family taken to be 1)

Null Deviance: 1996.506 on 1459 degrees of freedom

Residual Deviance: 1537.122 on 1379 degrees of freedom

BIC: 5175.649

MODEL WITH HIDDEN SKILL

Call: glm(formula = tutor$response ~ tutor$student + tutor$arithmatic + tutor$comprehend.one.step + tutor$composing.concrete.number.in.head + tutor$articulating.one.step + tutor$articulating.variable + tutor$articulating.composed.expression + tutor$sko.arithmatic + tutor$sko.comprehend.one.step + tutor$sko.composing.concrete.number.in.head + tutor$sko.articulating.one.step + tutor$sko.articulating.variable + tutor$sko.articulating.composed.expression, family = binomial)

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.634817	-0.8975559	0.4329174	0.8610784	2.83123

Coefficients: (3 not defined because of singularities)

	Value	Std. Error	t value
(Intercept)	4.485499363	0.45640902	9.82780617
tutor$student1	-0.662384317	0.35532869	-1.86414534
tutor$student2	0.338931399	0.21030904	1.61158742
tutor$student3	-0.353509192	0.20035366	-1.76442595
tutor$student4	0.417585081	0.14081090	2.96557351
tutor$student5	0.126170729	0.13129734	0.96095416
tutor$student6	0.029560099	0.10117779	0.29215996
tutor$student7	-0.043161213	0.14283459	-0.30217621
tutor$student8	0.079571753	0.07110622	1.11905476
tutor$student9	-0.061699293	0.04770284	-1.29340909
tutor$student10	-0.086307245	0.12376650	-0.69733931
tutor$student11	0.113131199	0.05756413	1.96530736
tutor$student12	0.081799915	0.04612601	1.77340094
tutor$student13	0.141534525	0.05730979	2.46963953
tutor$student14	0.016471885	0.03608582	0.45646415
tutor$student15	-0.001958697	0.04050127	-0.04836138
tutor$student16	0.063699670	0.03415494	1.86502047
tutor$student17	0.047869264	0.03268425	1.46459744
tutor$student18	0.001495270	0.02557240	0.05847201
tutor$student19	-0.031655045	0.02556072	-1.23842555
tutor$student20	-0.004445246	0.02208142	-0.20131161
tutor$student21	-0.022637036	0.05262569	-0.43015185
tutor$student22	0.104527212	0.03103224	3.36834224
tutor$student23	-0.012137281	0.02114455	-0.57401472
tutor$student24	0.030885087	0.02281531	1.35370018
tutor$student25	0.032048343	0.02459882	1.30284088
tutor$student26	-0.032593754	0.02430020	-1.34129560
tutor$student27	0.024449157	0.02258926	1.08233551
tutor$student28	0.044626600	0.02206495	2.02251040
tutor$student29	0.0467742674	0.020005122	2.33811457
tutor$student30	0.0104874071	0.016223026	0.64645196
tutor$student31	0.0453458547	0.020981532	2.16122711
tutor$student32	0.0455610837	0.022872997	1.99191579
tutor$student33	0.0214110385	0.016354669	1.30916976
tutor$student34	0.0376661205	0.018505670	2.03538273
tutor$student35	0.0203912763	0.016688167	1.22190031
tutor$student36	-0.0197299182	0.014631613	-1.34844450
tutor$student37	-0.0011344384	0.013606402	-0.08337534
tutor$student38	0.0051104895	0.022372981	0.22842238
tutor$student39	0.0404845602	0.018066519	2.24086107
tutor$student40	-0.0007528225	0.012977411	-0.05801022
tutor$student41	-0.0272867978	0.011961110	-2.28129302
tutor$student42	0.0136244760	0.014376431	0.94769530

	Value	Std. Error	t value
tutor$student43	-0.0065850829	0.014997315	-0.43908411
tutor$student44	-0.0089381095	0.010402724	-0.85920854
tutor$student45	0.0249559347	0.015432071	1.61714753
tutor$student46	0.0317131207	0.013660506	2.32151868
tutor$student47	-0.0008769156	0.014169035	-0.06188958
tutor$student48	0.0178045492	0.020370746	0.87402540
tutor$student49	0.0350084994	0.016984292	2.06122804
tutor$student50	-0.0176935260	0.012290086	-1.43965842
tutor$student51	0.0016440965	0.009510175	0.17287762
tutor$student52	-0.0148551593	0.009359317	-1.58720551
tutor$student53	0.0264076890	0.010628250	2.48466950
tutor$student54	0.0014023795	0.011488711	0.12206587
tutor$student55	0.0036274019	0.011232752	0.32293084
tutor$student56	-0.0119333196	0.008263562	-1.44408904
tutor$student57	-0.0301899321	0.009233742	-3.26952291
tutor$student58	0.1152008539	0.072228732	1.59494498
tutor$student59	-0.0176450488	0.010625486	-1.66063447
tutor$student60	-0.0307616141	0.006976959	-4.40902863
tutor$student61	-0.0006584258	0.010853230	-0.06066634
tutor$student62	0.0093051008	0.011829537	0.78659890
tutor$student63	0.0213113757	0.011385503	1.87179924
tutor$student64	-0.0171718402	0.007980946	-2.15160462
tutor$student65	-0.0152508545	0.006746469	-2.26056839
tutor$student66	-0.0273208903	0.007139457	-3.82674615
tutor$student67	0.0183665115	0.010951153	1.67713044
tutor$student68	-0.0002643060	0.007892609	-0.03348779
tutor$student69	-0.0255738486	0.009790496	-2.61210970
tutor$student70	-0.0087944048	0.009988823	-0.88042453
tutor$student71	0.0116237112	0.008755259	1.32762624
tutor$student72	-0.0028057063	0.007253656	-0.38679893
tutor$arithmatic	-3.3219527917	0.448577034	-7.40553471
tutor$comprehend.one.step	-2.4249093008	0.258189995	-9.39195690
tutor$composing.concrete.number.in.head	3.5764414851	0.895378592	3.99433437
tutor$articulating.one.step	NA	NA	NA
tutor$articulating.variable	-0.5638963423	0.283444737	-1.98944016
tutor$articulating.composed.expression	NA	NA	NA
tutor$sko.arithmatic	-0.5224687091	0.298756126	-1.74881338
tutor$sko.comprehend.one.step	0.3306102317	0.154183104	2.14427018
tutor$sko.composing.concrete.number.in.head	0.3525526102	0.600556153	0.58704354
tutor$sko.articulating.one.step	NA	NA	NA
tutor$sko.articulating.variable	-0.2542214091	0.153352735	-1.65775595
tutor$sko.articulating.composed.expression	-0.3391281476	0.164860116	-2.05706604

(Dispersion Parameter for Binomial family taken to be 1)
Null Deviance: 1996.506 on 1459 degrees of freedom
Residual Deviance: 1532.853 on 1378 degrees of freedom
BIC: 5181.671