Why Are Algebra Word Problems Difficult? Using Tutorial Log Files and the Power Law of Learning to Select the Best Fitting Cognitive Model

Ethan A. Croteau[1], Neil T. Heffernan[1], and Kenneth R. Koedinger[2]

[1] Computer Science Department
Worcester Polytechnic Institute
Worcester, MA. 01609, USA
{ecroteau, nth}@wpi.edu
[2] School of Computer Science
Carnegie Mellon University
Pittsburgh, PA. 15213, USA
koedinger@cmu.edu

Abstract. Some researchers have argued that algebra word problems are difficult for students because they have difficulty in comprehending English. Others have argued that because algebra is a generalization of arithmetic, and generalization is hard, it's the use of variables, per se, that cause difficulty for students. Heffernan and Koedinger [9] [10] presented evidence against both of these hypotheses. In this paper we present how to use tutorial log files from an intelligent tutoring system to try to contribute to answering such questions. We take advantage of the Power Law of Learning, which predicts that error rates should fit a power function, to try to find the best fitting mathematical model that predicts whether a student will get a question correct. We decompose the question of "Why are Algebra Word Problems Difficult?" into two pieces. First, is there evidence for the existence of this articulation skill that Heffernan and Koedinger argued for? Secondly, is there evidence for the existence of the skill of "composed articulation" as the best way to model the "composition effect" that Heffernan and Koedinger discovered?

1 Introduction

Many researchers had argued that students have difficulty with algebra word-problem *symbolization* (writing algebra expressions) because they have trouble comprehending the words in an algebra word problem. For instance, Nathan, Kintsch, & Young [14] "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 [9] [10] showed that many students can do *compute* tasks well, whereas they have great difficulty with the *symbolization* tasks [See Table 1 for examples of *compute* and *symbolization* types of questions]. They showed that many students could comprehend the words in the problem, yet still could not do the symbolization. An alternative explanation for "Why Are Algebra Word Problems Difficult?" is that the key is the use of

J.C. Lester et al. (Eds.): ITS 2004, LNCS 3220, pp. 240–250, 2004.

variables. Because algebra is a generalization of arithmetic, and it's the variables that allow for this generalization, it seems to make sense that it's the variables that make algebra symbolization hard.

However, Heffernan & Koedinger presented evidence that cast doubt on this as an important explanation. They showed there is hardly any difference between students' performance on *articulation* (see Table 1 for an example) versus *symbolization* tasks, arguing against the idea that the hard part is the presence of the variable per se.

Instead, Heffernan & Koedinger hypothesized that a key difficulty for students was in articulating arithmetic in the "foreign" language of algebra. They hypothesized the existence of a skill for articulating one step in an algebra word problem. This articulation step requires that a student be able to say (or "articulate") how it is they would do a computation, without having to actually do the arithmetic. Surprising, the found that is was easier for a student to actually do the arithmetic then to articulate what they did in an expression. To successfully articulate a student has to be able to write in the language of algebra. Question 1 for this paper is "Is there evidence from tutorial log files that support the conjecture that the *articulate* skill really exists?"

In addition to conjecturing the existence of the skill for articulating a single step, Heffernan & Koedinger also reported what they called the "composition effect" which we will also try to model. Heffernan & Koedinger took problems requiring two mathematical steps and made two new questions, where each question assessed each of the steps independently. They found that the difficulty of the one two-operator problem was much more than the combined difficulty of the two one-operator problems taken together. They termed this the composition effect. This led them to speculate as to what the "hidden" difficulty was for students that explained this difference in performance. They argued that the hidden difficulty included knowledge of *composition of articulation*. Heffernan & Koedinger attempted to argue that the composition effect was due to difficulties in articulating rather than on the task of comprehending, or at the symbolization step when a variable is called for. In this paper we will compare these hypotheses to try to determine the source of the composition effect originates. We refer to this as Question 2.

Heffernan & Koedinger's arguments were based upon two different samplings of about 70 students. Students' performances on different types of items were analyzed. Students were not learning during the assessment so there was no need to model learning. Heffernan & Koedinger went on to create an intelligent tutoring system, "Ms Lindquist", to teach student how to do similar problems. In this paper we attempt to use tutorial log file data collected from this tutor to shed light on this controversy. The technique we present is useful for intelligent tutoring system designers as it shows a way to use log file data to refine the mathematical models we use in predicting whether a student will get an item correct. For instance, Corbett and Anderson describe how to use "knowledge tracing" to track students performance on items related to a particular skill, but all such work is based upon the idea that you know what skills are involved already. But in this case there is controversy [15] over what are the important skills (or more generally, *knowledge components*). Because Ms Lindquist selects problems in a curriculum section randomly, we can learn what the knowledge components are that are being learned. With out problem randomization

we would have no hope of separating out the effect of problem ordering with the difficulty of individual questions.

In the following sections of this paper we present the investigations we did to look into the existence of both the skills of *articulation* as well as *composition of articulation*. In particular, we present mathematically predictive models of a student's chance of getting a question correct. It should be noted, such predicative models have many other uses for intelligent tutoring systems, so this methodology has many uses.

1.1 Knowledge Components and Transfer Models

As we said in the introduction, some [14] believed that comprehension was the main difficulty in solving algebra word problems. We summarize this viewpoint with our three skill transfer model that we refer to as the "Base" model.

The *Base Model* consists of arithmetic knowledge component (KC), comprehension KC, and using a variable KC. The transfer model indicates the number of times a particular KC has been applied for a given question type. For a two-step "compute" problem the student will have to *comprehend* two different parts of the word problem (including but not limited to, figuring out what operators to use with which literals mentioned in the problem) as well as using the *arithmetic* KC twice. This model can predict that symbolization problems will be harder than the articulation problems due to the presence of a variable in the symbolization problems. The *Base Model* suggests that computation problems should be easier than articulation problems, unless students have a difficult time doing arithmetic.

The KC referred to as "*articulating one-step*" is the KC that Heffernan & Koedinger [9] [10] conjectured was important to understanding what make algebra problems so difficult for students. We want to build a mathematical model with the *Base Model* KCs and compare it what we call the "*Base+Model"*, that also includes the *articulating one-step* KC.

So Question 1 in this paper compares the *Base Model* with a model that adds in the *articulating one-step* KC. Question 2 goes on to try to see what is the best way of adding knowledge components that would allow the model to predict the composition effect. Is the composition during the articulation, comprehension, articulation, or the symbolization? Heffernan and Koedinger speculated that there was a composition effect during articulation, suggesting that knowing how to treat an expression the same way you treat a number would be a skills that students would have to learn if they were to be good at problems that involved two-step articulation problems. If Heffernan & Koedinger's conjecture was correct, we would expect to find that the *composition of articulation* KC is better (in combination with one of the two Base Model variants) at predicting students difficulties than any of the other composition KCs.

1.2 Understanding How We Use This Model to Predict Transfer

Qualitatively, we can see that a our transfer model predicts that practice on one-step computation questions should transfer to one-step articulation problems only to the

degree that a student learns (i.e., receives practice at employing) the *comprehending one-step* KC. We can turn this qualitative observation into a quantified prediction method by treating each knowledge component as having a *difficulty* parameter and a *learning* parameter. This is where we take advantage of the Power Law of Learning, which is one of the most robust findings in cognitive psychology. The power law says that the performance of cognitive skills improve approximately as a power function of practice [16] [1]. This has been applied to both error rates as well as time to complete a task, but our use here will be with error rates. This can be stated mathematical as follows:

$$\text{Error Rate}(x) = b*x^{-d} \quad (1)$$

Where x represents the number of times the student has received feedback on the task, b represents a *difficulty* parameter related to the error rate on the first trail of the task, and d represents a *learning* parameter related to the learning rate for the task. Tasks that have large b values represent tasks that are difficult for students the first time they try it (could be due to the newness of the task, or the inherit complexity of the task). Tasks that have a large d coefficient represent tasks where student learning is fast. Conversely, small values of d are related to tasks that students are slow to improve[1].

The approach taken here is a variation of "learning factors analysis", a semi-automated method for using learning curve data to refine cognitive models [12]. In this work, we follow Junker, Koedinger, & Trottini [11] in using logistic regression to try to predict whether a student will get a question correct, based upon both item factors (like what knowledge components are used for a given question, which is what we are calling *difficulty* parameters), student factors (like a students pretest score) and factors that depend on both students and items (like how many times this particular students has practiced their particular knowledge component, which is what we are calling *learning* parameters.) Corbett & Anderson [3], Corbett, Anderson & O'Brien [4] and Draney, Pirolli, & Wilson [5] report results using the same and/or similar methods as described above. There is also a great deal of related work in the psychometric literature related to item response theory [6], but most of it is focused on analyzing test (e.g., SAT or GRE) rather than student learning.

1.3 Using the Transfer Model to Predict Transfer in Tutorial Log Files

Heffernan [7] created Ms. Lindquist, an intelligent tutoring system, and put it online (***www.algebratutor.org***) and collected tutorial log files for all the students learning to symbolize. For this research we selected a data set for which Heffernan [8] had previously reported evidence that students were learning during the tutoring sessions. Some 73 students were brought to a computer lab to work with Ms. Lindquist for two class periods totaling an average of about 1 hour of time for each student. We present

[1] All *learning* parameters are restricted to be positive otherwise the parameters would be modeling some sort of forgetting effect.

data from students working only on the second curriculum section, since the first curriculum was too easy for students and showed no learning. (An example of this dialog is shown in Table 2 and will be discussed shortly). This resulted in a set of log files from 43 students, comprising 777 rows where each row represents a student's first attempt to answer a given question.

Table 1. Showing a made-up tutor log file and how it uses the *Base+Model* Transfer Model

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
						Example Question	**Outcomes**		**Knowledge Components: *Difficulty* Parameters**				**Knowledge Components: *Learning* Parameters**			
	Scenario Identifier	**Factor:** *Task directions*	**Factor:** *Steps*	**Attempt at question type**	**First Attempt in sequence**		**Response Time (Seconds)**	**Response (Correct or not)**	**Arithmetic**	**Comprehending One-Step**	**Using a Variable**	**Articulating One-Step**	**Arithmetic**	**Comprehending One-Step**	**Using a Variable**	**Articulating One-Step**
1	1	S	2	1	T	**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**	**1**	**2**	**0**	**0**	**0**	**0**
2						**Student**:8-40/m	**130**	**N**								
3	1	C	2	1	T	**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?						**0**	**0**	**0**	**0**	**0**
4						**Student**:580	**38**	**N**								
5	1	C	1	1	T	**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**	**0**	**0**
6						**Student**:12	**19**	**N**								
7	1	C	1	2	F	**Tutor**: Hmm. No. Try again please. Please compute the distance rowed so far if the number of minutes rowing			**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**

						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.										
8						**Student**: 120	**9**	**Y**								
9	1	C	2	2	T	**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~~ 1**	**~~2~~ 1**	**0**	**0**	**1**	**1**	**0**	**0**
10						**Student**: 680	**24**	**Y**								
11	1	A	2	1	T	**Tutor**: Correct. Using only 800, 40, and 3 write an expression for the distance Anne has left to row			**0**	**~~2~~ 0**	**0**	**2**	**~~3~~ 2**	**~~3~~ 2**	**0**	**0**
12						**Student:** 120-800	**18**	**N**								
13	1	A	1	1	T	**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**	**0**	**1**	**~~3~~ 2**	**~~3~~ 2**	**0**	**0**
14						**Student:** 3*40	**22**	**Y**								
15	1	A	2	2	T	**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.				**~~2~~ 0**	**0**	**~~2~~ 1**	**~~3~~ 2**	**~~4~~ 2**	**0**	**1**
16						**Student**: 800-40*3	**18**	**Y**								
17	1	S	2	2	T	**Tutor**: Correct. Now use "m" instead of 3 to again express the distance Anne has left to row.			**0**	**~~2~~ 0**	**1**	**~~2~~ 0**	**~~3~~ 2**	**~~6~~ 2**	**0**	**~~3~~ 2**
18						**Student**:800-40*m	**9**	**Y**								
19						**Tutor**: Correct. Lets do a new problem.										
20	2	S	2	1	T	**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**	**1**	**2**	**~~3~~ 2**	**~~8~~ 2**	**1**	**~~5~~ 2**

Table 1 shows an example of the sort of dialog Ms. Lindquist carries on with students (this is with "made-up" student responses). Table 1 starts by showing a student working on scenario identifier #1 (Column 1) and only in the last row (Row 20) does the scenario identifier switch. Each word-problem has a single *top-level* question which is always a *symbolize* question. If the student 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 the problem down into simpler questions. The

combination of the second and third column indicates the question type. The second column is for the *Task Direction* factor, where S=Symbolize, C=Compute and A=Articulate. By crossing *task direction* and *steps,* there are six different question types. The 4th column defines what we call the *attempt* at a question type. The number appearing in the attempt column is the number of times the problem type has been presented during the scenario. For example, the first time one of the six question types is asked, the attempt for that question will be "1". Notice how on row 7, the attempt is "2" because it's the second time a *one-step compute* question has been asked for that scenario identifier. For another example see rows 3 and 7. Also notice that on line 20 the attempt column indicates a first attempt at a two-step symbolize problem for the new scenario identifier.

Notice that on row 5 and 7, the same question is asked twice. If the student did not get the problem correct at line 7, Ms Lindquist would have given a further hint of presenting six possible choices for the answer. For our modeling purposes, we will ignore the exact number of attempts the student had to make at any given question. Only the first attempt *in a sequence* will be included in the data set. For example, this is indicated in Table 1, in the 7th row of the 5th column, where the "F" for false indicates that row will be excluded from the data set.

The 6th column has the exact dialog that the student and tutor had. The 7th and 8th columns are grouped together because they are both outcomes that we will try to predict.[2] Columns 9-16 show what statisticians call the *design matrix*, which maps the possible observations onto the fixed effect (independent) coefficients. Each of these columns will get a coefficient in the logistic regression. Columns 9-12 show the *difficulty* parameters, while columns 13-16 show the *learning* parameters. We only list the four knowledge components of the *Base+ Model*, and leave out the four different ways to deal with composition. The *difficulty* parameters are simply the knowledge components identified in the transfer model. The *learning* parameter is calculated by counting the number of previous attempts a particular knowledge component has been learned (we assume learning occurs each time the system gives feedback on a correct answer). Notice that these learning parameters are strictly increasing as we move down the table, indicating that students' performance should be monotonically increasing.

Notice that the question asked of the student on row 3 is the same as the one on row 9, yet the problem is easier to answer after the system has given feedback on "the distance rowed is 120". Therefore the difficulty parameters are adjusted in row 9, column 9 and 10, to reflect the fact that if the student had already received positive feedback on those knowledge components. By using this technique we make the credit-blame assignment problem easier for the logistic regression because the number of knowledge components that could be blamed for a wrong answer had been reduced. Notice that because of this method with the difficulty parameters, we also had to adjust the learning parameters, as shown by the crossed out learning parame-

[2] 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.

ters. Notice that the learning parameters are **not** reset on line 20 when a new scenario was started because the learning parameters extend across all the problems a student does.

1.4 How the Logistic Regression Was Applied

With some minor changes, Table 1 shows a snippet of what the data set looked like that we sent to the statistical package to perform the logistic regression. We performed a logistic regressions predicting the dependent variable *response* (column 8) based on the independent variables on the knowledge components (i.e., columns 9-16). For some of the results we present, we also add a student specific column (we used a student's pretest score) to help control for the variability due to students differing incoming knowledge.

2 Procedure for the Stepwise Removal of Model Parameters

This section discusses how a fit model is made parsimonious by a stepwise elimination of extraneous coefficients. We only wanted to include in our models those variables that were reasonable and statistically significant. The first criterion of reasonableness was used to exclude a model that had "negative" learning curves that predict students would do worse over time. The second criterion of being statistically significant was used to remove, in a stepwise manner, coefficients that were not statistically significant (those coefficients with t-values between 2 and –2 is a rule of thumb used for this). We choose, somewhat arbitrarily, to first remove the learning parameters before looking at the difficulty parameters. We made this choice because the learning parameters seemed to be, possibly, more contentious. At each step, we chose to remove the parameter that had the least significance (i.e., the smallest absolute t-value).

A systematic approach to evaluating a model's performance (in terms of error rate) is essential to comparing how well several models built from a training set would perform on an independent test set.

We used two different was of evaluating the resulting models: BIC and a k-holdout strategy. The Bayesian Information Criterion is one method that is used for model selection [17] that tries to balance goodness of fit with the number of parameters used in the model. Intuitively, BIC, penalizes models that have more parameters. Differences in BIC greater than 6 between models are said to be strong evidence while differences of greater than 10 is said to be very strong (See [2] for another example of cognitive model selection using BIC for model selection in this way.)

We also used a k-holdout strategy that worked as follows. The standard way of predicting the error rate of a model given a single, fixed sample is to use a stratified k-fold cross-validation (we choose k=10). Stratification is simply the process of randomly selecting the instances used for training and testing. Because the model we are trying to build makes use of a student's successive attempts, it seemed sensible to randomly select whole students rather than individual instances. Ten fold implies the training and testing procedure occurs ten times. The stratification process created a

testing set by randomly selecting one-tenth of the students not having appeared in a prior testing set. This procedure was repeated ten times in order to have included each student in a testing set exactly once.

A model was then constructed for each of the training sets using a logistic regression with the student response as the dependent variable. Each fitted model was used to predict the student response on the corresponding testing set. The prediction for each instance can be interpreted as the model's fit probability that a student's response was correct (indicated by a "1"). To associate the classification with the bivariate class attribute, the prediction was rounded up or down depending if it was greater or less than 0.5. The predictions were then compared to the actual response and the total number of correctly classified instances were divided by the total number of instances to determine the overall classification accuracy for that particular testing set.

3 Results

We summarize the results of our model construction, with Table 2 showing the results of models we attempted to construct. To answer Question 1, we compared the *Base Model* to the *Base+ Model* that added the *articulate one-step* KC. After applying our criterion for eliminating non-statistically significant parameters we were left with just two difficulty parameters for the *Base Model* (all models in Table 2 also had the very statistically significant pretest parameter).

Table 2. Models Computed: BIC and K-holdout evaluation, and the KC in each unique model

	Models	
Name	Base	Base +
Model #	0	1
BIC	2508.9	2493.7
Overall Evaluation	59.6%	64.3%
KCs	Comprehending one-step	Articulating-one-step
	Articulating variable	Articulating variable
		Arithmetic

It turned out that the *Base+ Model* did a better statistically significant better job (smaller BIC are better) than the *Base Model* in terms of BIC (the difference was great than 10 BIC points suggesting a statistically significant difference). The *Base+ Model* also did better when using the K-holdout strategy (59.6% vs 64.3%). We see from Table 2 that the *Base+ Model* eliminated the *comprehending one-step* KC and added instead the *articulating one-step* and *arithmetic* KCs suggesting that "articulating" does a better job than comprehension as the way to model what is hard about word problems.

So after concluding that there was good evidence for *articulating one-step*, we then computed Models 2-4. We found that two of the four ways of trying to model composition resulted in models that were inferior in terms of BIC and not much different in terms of the K-holdout strategies. We found that models 4 and 5 were reduced to the *Base+ Model* by the step-wise elimination procedure. We also tried to calculate the effect of combining any two of the four composition KCs but all such attempts were reduced by the step-wise elimination procedure to already found models. This suggests that for the set of tutorial log files we used, there was not sufficient evidence to argue for the *composition of articulation* over other ways of modeling the composition effect.

It should be noted that while none of the learning parameters of any of the knowledge components were in any of the final models (thus creating models that predict no learning over time) we should note that on models 4 and 5, the last parameter that was eliminated was a *learning* parameters that both had t-test values that were within a very small margin of being statistically significant (t=1.97 and t=1.84). It should also be noted that in Heffernan [8] the learning within Experiment 3 was only close to being statistically significant. That might explain why we do not find any statistically significant learning parameters.

We feel that Question 1 ("Is there evidence from tutorial log files that support the conjecture that the *articulating one-step* KC really exists?") is answered in the affirmative, but Question 2 ("What is the best way to model the composition effect") has not been answered definitely either way. All of the models that tried to explicitly model a composition KC did not lead to significantly better models. So it is still an open question of how to best model the composition effect.

4 Conclusions

This paper presented a methodology for evaluating models of transfer. Using this methodology we have been able to compare different plausible models. We think that this method of constructing transfer models and checking for parsimonious models against student data is a powerful tool for building cognitive models.

A limitation of this techniques is that the results depend on what curriculum (i.e., the problems presented to students, and the order in which that happened) the students were presented with during their course of study. If students were presented with a different sequence of problems, then there is no guarantee of being able to draw the same conclusions.

We think that using transfer models could be an important tool to use in building and designing cognitive models, particularly where learning and transfer are of interest. We think that this methodology makes a few reasonable assumptions (the most important being the Power Law of Learning). We think the results in this paper show that this methodology could be used to answer interesting cognitive science questions.

References

1. Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought.* Lawrence Erlbaum Associates, Mahwah, NJ.
2. 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.
3. Corbett, A. T. and Anderson, J. A. (1992) Knowledge tracing in the ACT programming tutor. In: Proceedings of 14-th Annual Conference of the Cognitive Science Society.
4. 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.
5. 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.
6. Embretson, S. E. & Reise, S. P. (2000) Item Response Theory for Psychologists Lawrence Erlbaum Assoc.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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
12. Koedinger, K. R. & Junker, B. (1999). Learning Factors Analysis: Mining student-tutor interactions to optimize instruction. Presented at Social Science Data Infrastructure Conference. New York University. November, 12-13, 1999.
13. 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.
14. 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.
15. Nathan, M. J., & Koedinger, K. R. (2000). Teachers' and researchers' beliefs about the development of algebraic reasoning. *Journal for Research in Mathematics Education*, 31, 168-190.
16. Newell, A., & Rosenbloom, P. (1981) Mechanisms of skill acquisition and the law of practice. In Anderson (ed.), *Cognitive Skills and Their Acquisition*., Hillsdale, NJ: Erlbaum.
17. Raftery, A.E. (1995) Bayesian model selection in social research. *Sociological Methodology* (Peter V. Marsden, ed.), Cambridge, Mass.: Blackwells, pp. 111-196 .