Does help help? Introducing the Bayesian Evaluation and Assessment methodology

Joseph E. Beck[1], Kai-min Chang[2], Jack Mostow[2], and Albert Corbett[2]
[1]Computer Science Department, Worcester Polytechnic Institute
[2]School of Computer Science, Carnegie Mellon University
joseph.beck@EducationalDataMining.org
{mostow, kkchang, corbett}@cs.cmu.edu

Abstract. Most ITS have a means of providing assistance to the student, either on student request or when the tutor determines it would be effective. Presumably, such assistance is included by the ITS designers since they feel it benefits the students. However, whether—and how—help helps students has not been a well studied problem in the ITS community. In this paper we present three approaches for evaluating the efficacy of the Reading Tutor's help: creating experimental trials from data, learning decomposition, and Bayesian Evaluation and Assessment, an approach that uses dynamic Bayesian networks. We have found that experimental trials and learning decomposition both find a negative benefit for help--that is, help hurts! However, the Bayesian Evaluation and Assessment framework finds that help both promotes student long-term learning and provides additional scaffolding on the current problem. We discuss why these approaches give divergent results, and suggest that the Bayesian Evaluation and Assessment framework is the strongest of the three. In addition to introducing Bayesian Evaluation and Assessment, a method for simultaneously assessing students and evaluating tutorial interventions, this paper describes how help can both scaffold the current problem attempt as well as teach the student knowledge that will transfer to later problems.

Key words: educational data mining, dynamic Bayesian networks, assessment, evaluation, Bayesian Evaluation and Assessment

1 Introduction

An important property of an Intelligent Tutoring System (ITS) is its ability to help students. Thus, measuring the effectiveness of tutor help is an important evaluation of an ITS. Does help help? Does one type of help work better than the others [1]? Even though the tentative answer is "yes" by most ITS researchers (otherwise, why include help at all in the tutor?), answering such questions is surprisingly difficult. One of the difficulties is that the question "does help help?" is ill-defined; what does it mean to help students? Does it mean to assist students in performing correctly on the current attempt or does it mean to assist in learning of persistent knowledge that will help on future attempts?

To measure the effectiveness of tutor help, we would ideally set up a controlled pre- and post- test experiment. A typical experimental setup works as follows: in the pre-test, we assess student performance before using the ITS. Then, we randomly assign students into two groups. The experimental group uses one

version of ITS *with* the tutor help that we're evaluating, whereas the control group uses another version of ITS *without* the particular tutor help. After students use the ITS for some time, we assess student performance of the two groups again in the post-test. We test the hypothesis that the performance improvement in the experimental group is significantly different than in the control group. This experimental design is sound and has been extensively practiced in the field of psychology. Nonetheless, the experimental design is often impractical for evaluating an ITS because a controlled experiment takes a long time to conduct and is often too expensive to conduct, although exceptions exist [2].

Given that the ideal pre- and post-test experimental studies are often impractical, there are several other approaches to measure the effectiveness of tutor help. For example, we may conduct user case studies and directly ask the students whether they find the tutor help effective. Unfortunately, while user case studies provide valuable qualitative feedback, they lack the ability to draw conclusive relationships. Alternatively, we can try to infer tutor help efficacy *from data*. For instance, one might claim that tutor help is effective if student performance improves when they receive help, compared to when they do not receive help. However, this approach raises the question of *when* to assess student performance. Immediate performance is prone to scaffolding effects where tutor help merely provides a short-term performance boost. For example, some help types provide students the answer; if students simply imitate the answer we should not count that as learning.

In this paper, we describe a methodology to model both tutor help and student knowledge in one coherent framework. This configuration allows us to tease apart the effect of help into 1) scaffolding immediate performance and 2) teaching persistent knowledge that improves long term performance. We evaluate the proposed framework on student performance data from the Reading Tutor, an Intelligent Tutoring System that listens to children read aloud [3]. The Reading Tutor uses automated speech recognition to listen to children read aloud and tries to score their reading as correct or incorrect. Students can ask for assistance on a challenging word, and the Reading Tutor chooses randomly which type of help to give. For example, if the student clicks on the word "cat" the tutor could say "Rhymes with…bat"; it could sound out the word, break longer words into syllables; or simply speak the word for the student. If the student does not like the help provided, he can click again and receive another random selection.

2 Naïve approaches to modeling help

There are several approaches one could apply to observational data to estimate the efficacy of the tutor's help. We first discuss experimental trials and learning decomposition.

2.1 Experimental trials

For experimental trials, two items are needed: what is being compared, and the outcome measure with which to perform the comparison. Note that all of the scoring in this paper is performed by automated speech recognition (ASR), which is not nearly as accurate as typed input. Therefore, interpreting a number in isolation or numbers derived from a small number of observations is suspect. In terms of what is

being compared, the issue is somewhat problematic. One natural item of interest is student performance on words on which he clicked for help. One possible comparison is words on which he did not ask for help.

In terms of an outcome measure, student performance on the word is a natural measure to use. Since students are reading stories aloud to the Reading Tutor, we expect students to periodically encounter words simply in the course of reading, and we can use those as our outcome. If we use student performance at reading the word immediately after receiving help, we find that words on which the student receives help are read more accurately than words on which he does not. However, this outcome is contaminated by recency effects. For example, if tutor read *antidisestablishmentarianism* aloud, and the student immediately mimicked the tutor, we should not necessarily be confident that he actually knows the word. Perhaps the pronunciation was simply in his working memory buffer [4].

A stronger outcome is one that avoids memory effects by waiting for a later day to test the student's performance. If we change the outcome to only consider cases where the student encounters the word on a later day, then we find that words that did not receive help were read with 83% accuracy, while words that received help were read with 73% accuracy. In other words, help is providing a "benefit" to students of 10% worse performance. Although the Reading Tutor's help could certainly be improved, we are skeptical that its assistance is *that* bad. A more likely explanation is that students click for help on words they do not know. If a student doesn't know a word, the help might help him to learn it, but even after receiving the help he probably will not understand the word as well as one that he already knew. Therefore, the difference in performance on later days is more a function of the student's starting knowledge than a function of receiving the help.

2.2 Learning decomposition

A slightly more sophisticated technique is learning decomposition [5-7]. Learning decomposition is a variant of learning curves. Typically learning curves estimate how much students learn as a result of a practice opportunity. Learning decomposition instead estimates the relative worth of different types of learning opportunities. For example, prior work with learning decomposition has shown that students learn approximately 25% more in stories they choose to read vs. those selected by the tutor [8]. It is possible to apply learning decomposition to this analysis by considering two types of learning encounters: reading words and receiving help. In this way, we can see how valuable help is compared to simply reading the word.

Unlike the experimental trials approach, it is not necessary to construct a comparison set of words. Learning decomposition simply computes the relative impact of help compared to reading the word. Similar to the experimental trials approach, it is necessary to decide what the set of allowable outcomes will be. Again, to avoid recency effects, we only consider words that students encounter on later days. We fit the model shown in Equation 1 to each student's data. By doing so, we get an estimate of the impact of help for each student, controlling for the fixed traits of the student (this control is analogous to that from having the student be a factor in logistic regression).

$$readingtime = A * e^{-b*(r*m*RM + r*RD + m*NM + ND + h*\#helps)}$$

Equation 1. Learning decomposition formula for evaluating the impact of help

The learning decomposition model finds that reading a word is, by definition, worth 1.0 practice opportunities. Relative to this baseline, depending on the exact model used, help is worth roughly -1.5 to -4 trials of learning (e.g. the model reported in [5] produces a result of -1.91, while the model shown in Equation 1 gives a result of -3.3 exposures). That is, receiving help caused students to perform worse on later trials compared to words on which they did not receive help. Even after controlling for student properties by constructing a per-student model, and comparing the effect of help relative to a baseline of simply reading a word, help is still found to be unhelpful. We suspect a similar effect is occurring is with the experimental trials approach: students request help on words on which they have low knowledge. The help thus acts as *evidence* of a lack of knowledge, rather than a direct *cause* of that lack of knowledge.

3 New approach: Bayesian Evaluation and Assessment

There are two primary failures with the above-mentioned naïve approaches to modeling the impact of help. First, controlling for students is not sufficient. Rather, it is necessary to control for the student's knowledge of the skill (in our case, word) being helped. Second, it is necessary to refine exactly what is meant by help helping the student. Although both of the prior analyses ignored the short-term impact of help on performance, that may not be the best approach. Students typically request help when they are stuck; if help can get them unstuck then it can be said to be at least partially effective. Such a temporary boost is of course no substitute for truly effective help, otherwise ITS designers would have help systems that simply told students the answer. However, there should at least be some acknowledgement of short term benefits.

To achieve these goals, we unify two common analysis goals in ITS. The first of these is *assessing* student knowledge. Most ITS have some form of assessment or student modeling. Figure 1 shows a graphical representation of knowledge tracing [8], a fairly common student modeling approach. This relatively simple dynamic Bayesian network suffices to completely describe knowledge tracing [9]. The shaded nodes represent things the model can directly observe, in this case student performance. The unshaded nodes represent unobservable latent parameters, in this case the student's knowledge. Each pair of knowledge and performance represents one practice opportunity for a particular skill. So in this example there are two practice opportunities represented.

The arrow from student knowledge to student performance indicates that knowledge influences performance. Performance is assumed to be a noisy reflection of knowledge, and is mediated by two parameters. The *guess* parameter represents that a student may sometimes generate a correct response in spite of not knowing the correct skill. The *slip* parameter acknowledges that even students who understand the skill can make an occasional careless mistake. The definition of each of the parameters in Figure 1 is shown in Equation 2. The link between student knowledge

across time slices indicates that students maintain and hopefully increase their knowledge across time slices. Although both learning and forgetting can occur in the real world, we follow standard practice for knowledge tracing and set the forgetting parameter to be 0.

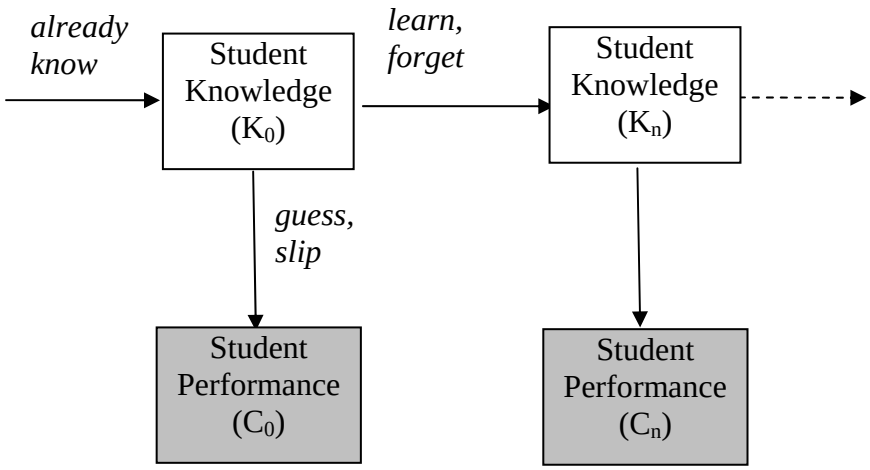

Figure 1. Diagram of knowledge tracing

.

$$
\begin{aligned}
\textit{already know} &\equiv \Pr(K_0 = \textit{true}) \\
\textit{learn} &\equiv \Pr(K_n = \textit{true} \mid K_{n-1} = \textit{false}) \\
\textit{forget} &\equiv \Pr(K_n = \textit{false} \mid K_{n-1} = \textit{true}) = 0 \\
\textit{guess} &\equiv \Pr(C_n = \textit{true} \mid K_n = \textit{false}) \\
\textit{slip} &\equiv \Pr(C_n = \textit{false} \mid K_n = \textit{true})
\end{aligned}
$$

Equation 2. Equations representing knowledge tracing parameters

The second common analysis goal in ITS is to evaluate tutorial interventions. Figure 2 shows graphically how such evaluations are frequently performed [e.g. 1, 10]. In this case, both the student performance and the intervention are observable. The approach is to determine how the intervention influences student performance. Since both nodes are observable, this task is typically easier than student modeling. Our approach is to combine these two methodologies, assessment and evaluation, into a single modeling framework, shown in Figure 3. The Student Knowledge and Student Performance nodes are similar to the ones in knowledge tracing. The new node represents a binary variable: did the tutorial intervention we are evaluating occur during this practice opportunity? This node creates two new arcs in the network. The first one, *teach*, connects the tutorial intervention with student knowledge. It models the impact the tutorial intervention could have on the student's knowledge. Note that this arc carries forward to future time slices, so the impact on the student will persist. The second new arc, *scaffold [11]*, represents the impact the intervention has on the

current practice opportunity. This temporary support does not persist across problems, and only serves to aid the student on the current problem.

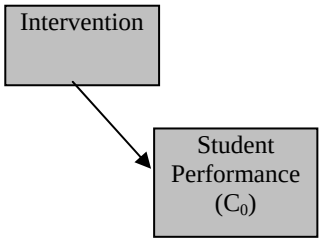

Figure 2. Common approach for evaluating tutorial interventions.

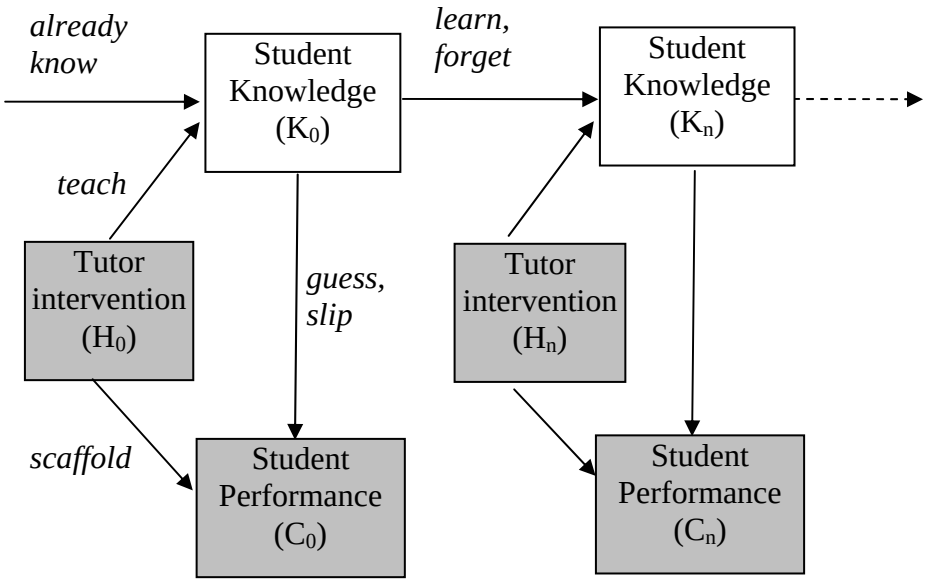

Figure 3. Bayesian Evaluation and Assessment architecture

This approach simultaneously assesses the student, since performance and knowledge are linked, and evaluates the tutorial intervention both in the context of its temporary benefit to student performance and its lasting impact on student knowledge. The impact of the tutorial intervention can be determined by examining the parameters learned by the model. For example P(learn | tutor intervention = false) is the baseline probability that a student will acquire a skill simply by practicing it. P(learn | tutor intervention = true) is the probability the student will acquire the skill as a result of receiving both the intervention and a chance to practice the skill. Comparing these two parameters permits us to estimate how much learning the intervention causes.

Similarly, the scaffolding effect on student performance can be estimated by comparing P(correct response | student didn't know the skill, intervention=false) vs. P(correct response | didn't know the skill, intervention=true). Any difference in performance is the scaffolding effect of the intervention. Equation 3 shows all of the equations for the Bayesian Evaluation and Assessment model.

$$already\ know \mid help \equiv \Pr(K_0 = true, \mid H_0 = true)$$
$$already\ know \mid no\ help \equiv \Pr(K_0 = true, \mid H_0 = false)$$

$$learn \mid help \equiv \Pr(K_n = true \mid K_{n-1} = false, H_n = true)$$
$$learn \mid no\ help \equiv \Pr(K_n = true \mid K_{n-1} = false, H_n = false)$$

$$forget \mid help \equiv \Pr(K_n = false \mid K_{n-1} = true, H_n = true)$$
$$forget \mid no\ help \equiv \Pr(K_n = false \mid K_{n-1} = true, H_n = false)$$

$$guess \mid help \equiv \Pr(C_n = true \mid K_n = false, H_n = true)$$
$$guess \mid no\ help \equiv \Pr(C_n = true \mid K_n = false, H_n = false)$$

$$slip \mid help \equiv \Pr(C_n = false \mid K_n = true, H_n = true)$$
$$slip \mid no\ help \equiv \Pr(C_n = false \mid K_n = true, H_n = false)$$

Equation 3. Equations for parameters in Bayesian Evaluation and Assessment model

4 Results

Our data came from 360 children between six and eight years old who used Project LISTEN's Reading Tutor [3] in the 2002-2003 school year. On average, students used the tutor for 8.5 hours. Over the course of the school year, these students read approximately 1.95 million words (as heard by the automatic speech recognizer). We separated the data into training and testing sets by splitting the students into two groups. We sorted the students according to their amount of Reading Tutor usage and then alternately assigned students to the two sets.

During a session with the Reading Tutor, the tutor presented one sentence (or fragment) at a time for the student to read aloud. The student's speech was segmented into utterances delimited by silences. Each utterance was processed by the Automatic Speech Recognizer (ASR) and aligned against the sentence. This alignment scored each word of the sentence as either accepted (classified by the ASR as read correctly) or rejected (thought to be misread or omitted). ASR acceptance is modeled as the observed performance (C_n). The tutorial intervention node is instantiated by whether the student received help on a word. For modeling purposes, this paper treats each English word as a separate skill.

We make use of a generic Bayes net toolkit for student modeling, BNT-SM [12], for our experiments. BNT-SM inputs a data set and a compact XML specification of a DBN model hypothesized by a researcher to describe causal

relationships among student knowledge and observed behavior. It generates and executes the code to train and test the model using the Bayes Net Toolbox [13]. BNT-SM allows researchers to easily explore different hypotheses on how is knowledge represented in a student model. We now show how to use BNT-SM to construct a DBN that models the effectiveness of tutor help on student knowledge.

We used the BNT-SM and Expectation Maximization (EM) algorithm to optimize data likelihood (i.e. the probability of observing our student performance data) in order to estimate our model's parameters. EM is the standard algorithm used in the machine learning community to estimate DBN parameters when the structure is known and there exist latent variables (e.g., student knowledge, K_n). EM is guaranteed to converge to a local maximum on the likelihood surface. We used the junction tree procedure for exact inference (estimating the value of the hidden variables). See Jensen [14] for a thorough introduction to Bayes nets and the standard training and inference algorithms.

4.1 Results for evaluating help

To evaluate the effectiveness of tutor help, we first compare the model parameters estimated by our Bayesian Evaluation and Assessment model with those obtained by estimating a simpler knowledge tracing model.

Table 1 shows the parameters estimated for the KT model and the Bayesian Evaluation and Assessment (Help, for short) model, respectively. Notice that the KT model does not consider the help information, whereas the Help model has the parameters conditioned on whether or not tutor help is given or not. As seen in the Help model of Table 1, the probability of *already know* (i.e. does the student know the word when first starting to use the tutor) is much higher when there is no help than when there is help. This suggests that tutor help is more likely to be provided for words the student is less likely to know– a positive finding. Also, the probability of *learning* is higher when there is help than when there is no help. Even though the effect of help is only an 8% relative improvement, it is at least in the right direction (unlike the two baseline approaches), suggesting that tutor help does have a positive effect on long term knowledge acquisition.

Table 1. Comparing the parameters estimated by the KT model and the Help model

	KT model	Help model	
		No Help Given	Help Given
Already know	0.618	0.660	0.278
Learn	0.077	0.083	0.088
Guess	0.689	0.655	0.944
Slip	0.056	0.058	0.009

Also as seen in Table 1, the probability of guess is higher when there is help than when there is no help and the probability of slip is higher when there is no help than when there is help. In other words, even if the student does not know the skill he is much more likely to generate a correct response when he receives than when he does not: 94% vs. 66% (the guess rate is inflated when applying knowledge tracing to student models that use speech recognition for scoring [15]). This finding suggests that tutor help does have a scaffolding effect on assisting immediate performance.

Notice that, although we have argued that teaching effect is more beneficial in the long run than the scaffolding effect, we cannot ignore the latter. For instance, if a student is stuck when using the tutor, the tutor should still help the student to become unstuck. Finally, both the teaching and scaffolding effects are statistically reliable at $p<0.05$ (paired samples t-test, done per-skill to avoid intraskill correlation), suggesting that tutor help does have an effect on both student knowledge and student performance.

4.2 Results for modeling students

Although the goal of this paper is to determine whether and how help helps students, our model also estimates student knowledge as a side effect. Therefore, we evaluate its performance at doing so. Since student knowledge is a latent variable that cannot be directly observed, we have no gold standard against which to compare. Instead, we used the trained student model to predict whether the ASR would accept or reject a student's next attempt to read the word. That is, we observe reading item by item and predict whether the next word will be read correctly (in not yet seen test data). An ROC (Receiver Operating Characteristic) curve measures the performance of a binary classifier by plotting the true positive rate against the false positive rate for varying decision thresholds. The area under the ROC curve (AUC) is a reasonable performance metric for classifier systems, assuming no knowledge of the true ratio of misclassification costs [16].

On our training data, the new model had near identical performance to classic knowledge tracing: AUC of 0.654 vs. 0.652. On the held out test data, performance was again a tie: AUC of 0.612 vs 0.615. It is disappointing our approach of simultaneously assessing students and evaluating the tutor did not yield more accurate assessment.

5 Contributions

This paper makes three main contributions to the ITS literature. First, the Bayesian Evaluation and Assessment framework unifies several strands of research. It is based on knowledge tracing [8] for assessing students. There has also been work on creating a node to measure the impact of help and connecting it to student knowledge [17]. However, this work used a simplified version of knowledge tracing, and was never evaluated with actual student data. The third strand is the ANDES system [18], which has a link between student knowledge and performance—that is, it assumes that help provides a scaffolding effect—but not between help and knowledge. Furthermore, the parameters in the ANDES system were not estimated from data.

The second contribution this paper makes is the conceptual one of simultaneously representing tutor interventions and the student's knowledge. Previous approaches addressed these problems separately by ignoring one to solve the other [1,3]. Specifically, KT ignored help, and some other experiments [3] ignored student knowledge, or how it changed over time.

The third contribution this paper makes is on distinguishing between two effects of help: scaffolding immediate performance vs. boosting actual learning. Prior work either assumed help has no direct impact on student learning [18] or that help has no direct impact on student performance [17]. Moreover, because we model tutor help and student knowledge in one coherent framework, we can estimate the

scaffolding and teaching effects. This separation of immediate vs. persistent effect of help allows researchers to understand what the tutor intervention is really doing. For instance, it is possible to investigate whether some tutor interventions help persistent learning while others mainly help immediate performance.

6 Future work

Currently, due to limitations in BNT-SM, we could only test models with discrete, binary variables. For example, in the Help model, we only answer the question "*does help help at all*?" A more interesting question to ask is "*which type of help helps more, and when is it effective?*" Thus, a future study is to extend BNT-SM to handle multinomial variables, which allows modeling of different help types.

One question that we are interested in exploring is how does our dynamic Bayes net framework compare to the pre- and post- test experimental design [2]. Do they draw similar conclusions, despite the fact that an experimental design is usually more expensive to conduct than data fitting with DBNs? Moreover, what kinds of causal relationship can we infer with our Bayesian framework? That we were able to get a positive result with a non-randomized intervention, whether a student receives help, suggests the framework should perform well at analyzing actual randomized controlled trials, and may even be a more sensitive measure due to its accounting for student knowledge.

Another issue that we are interested in addressing is to better understand why we cannot better model students in our new framework. One possible explanation is there are too many parameters. The impact of help is modeled independently for all 3000 skills (words) in the domain. Some way of simplifying the parameter search by using hierarachical models or Dirichlet priors [19] may be a solution.

Finally, we would like to conclude that our Bayesian Evaluation and Assessment architecture is the most accurate of the three approaches proposed in this paper. On balance, given that the other two approaches found that help is harmful, we can tentatively conclude that the new approach is better. However, better clarifying which approach is most accurate and when would be helpful. This question cannot be answered for interventions whose true effectiveness is unknown (i.e. all ITS interventions that exist in the real world). Therefore, evaluations with synthetic data [e.g. 17] are a promising route forward.

7 Conclusions

This paper presents a new approach, Bayesian Evaluation and Assessment, that we used to measure the impact of help. Of the three approaches we considered, our new framework gave the only plausible answer to the question "does help help?" Although the result is equivocal, as we do not know the "real" answer, it is important to note that this drawback is fundamental to **every** method of measuring an intervention's effectiveness. Typically when a number is presented purporting to represent how well an intervention worked, there is no discussion of alternate methods of doing the measuring. By putting the three numbers forward we acknowledge the problem, and argue that only one of the numbers is plausibly correct. The Reading Tutor's on-demand help is potentially useless, and we would not disregard the possibility, but the negative impacts claimed by the other two approaches are simply implausible.

The reason our new framework is superior is that it controls for student knowledge while estimating the intervention's effectiveness. Conceptually, this simultaneous modeling is similar to item response theory [20], which enables better comparisons of students across groups by simultaneously estimating student proficiency and question difficulty.

Finally, we feel it is important to enumerate both impacts of assistance: short term performance boosts (scaffolding) as well as longer term learning gains (teaching). By simultaneously addressing all of these aspects of assessment and evaluation, this framework represents a step forward in ITS evaluation methodology.

Acknowledgements

This work was supported by the National Science Foundation, ITR/IERI Grant No. REC-0326153. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation

References

1. Heiner, C., J.E. Beck, and J. Mostow. *Improving the help selection policy in a Reading Tutor that listens*. in *Proceedings of the InSTIL/ICALL Symposium on NLP and Speech Technologies in Advanced Language Learning Systems*. 2004. p. 195-198 Venice, Italy.
2. Arroyo, I., J.E. Beck, C.R. Beal, R.E. Wing, and B.P. Woolf. *Analyzing students' response to help provision in an elementary mathematics Intelligent Tutoring System*. in *Help Provision and Help Seeking in Interactive Learning Environments. Workshop at the Tenth International Conference on Artificial Intelligence in Education.* 2001. p. San Antonio, TX.
3. Mostow, J. and G. Aist, *Evaluating tutors that listen: An overview of Project LISTEN*, in *Smart Machines in Education*, P. Feltovich, Editor. 2001, MIT/AAAI Press: Menlo Park, CA. p. 169-234.
4. Anderson, J.R., *Rules of the mind*. 1993, Hillsdale, NJ: Lawrence Erlbaum Associates.
5. Beck, J. *Using learning decomposition to analyze student fluency development*. in *ITS2006 Educational Data Mining Workshop*. 2006. p. Jhongli, Taiwan.
6. Zhang, X., J. Mostow, and J.E. Beck. *All in the (word) family: Using learning decomposition to estimate transfer between skills in a Reading Tutor that listens*. in *AIED2007 Educational Data Mining Workshop*. 2007. p. 80-87.
7. Beck, J.E. *Does learner control affect learning?* in *Proceedings of the 13th International Conference on Artificial Intelligence in Education.* 2007. p. 135-142 Los Angeles, CA.
8. Corbett, A.T. and J.R. Anderson, *Knowledge tracing: Modeling the acquisition of procedural knowledge.* User Modeling and User-Adapted Interaction, 1995. **4**: p. 253-278.

9. Reye, J., *Student Modelling based on Belief Networks.* International Journal of Artificial Intelligence in Education, 2004. **14**: p. 1-33.
10. Mostow, J., J. Beck, J. Bey, A. Cuneo, J. Sison, B. Tobin, and J. Valeri, *Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial interventions.* Technology, Instruction, Cognition and Learning, 2004. **2**: p. 97-134.
11. Vygotsky, L., *Play and its role in the mental development of the child*, in *Play: Its role in development and evolution (1976)*, J. Bruner, A. Jolly, and K. Sylva, Editors. 1933, Penguin Books: New York. p. 461-463.
12. Chang, K.-m.K., J.E. Beck, J. Mostow, and A. Corbett. *A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems*. in *8th International Conference on Intelligent Tutoring Systems*. 2006. p. Jhongli, Taiwan.
13. Murphy, K., *Bayes Net Toolbox for Matlab*. 1998.
14. Jensen, F.V., ed. *Bayesian Networks and Decision Graphs.* Statistics for Engineering and Information Science, ed. M. Jordan, Lauritzen, S. L., Lawless, J. F., Nair, V. 2001, Springer.
15. Beck, J.E. and J. Sison, *Using knowledge tracing in a noisy environment to measure student reading proficiencies.* International Journal of Artificial Intelligence in Education, 2006. **16**: p. 129-143.
16. Hand, D., H. Mannila, and P. Smyth, *Principles of Data Mining.* 2001, Cambridge, Massachusetts: MIT Press.
17. Jonsson, A., J. Johns, H. Mehranian, et al. *Evaluating the Feasibility of Learning Student Models from Data*. in *Educational Data Mining: Papers from the AAAI Workshop*. 2005. p. 1-6 Pittsburgh: AAAI Press.
18. Conati, C., A. Gertner, and K. VanLehn, *Using Bayesian Networks to Manage Uncertainty in Student Modeling.* User Modeling and User-Adapted Interaction, 2002. **12**(4): p. 371-417.
19. Beck, J.E. and K.-m. Chang. *Identifiability: A Fundamental Problem of Student Modeling*. in *International Conference on User Modeling*. 2007. p. 137-146 Corfu, Greece.
20. Embretson, S.E. and S.P. Reise, *Item Response Theory for Psychologists*. Multivariate Applications, ed. L.L. Harlow. 2000, Mahwah: Lawrence Erlbaum Associates. 371.