Pardos, Z. A., Heffernan, N. T., Ruiz, C. & Beck, J. (2008) Effective Skill Assessment Using Expectation Maximization in a Multi Network Temporal Bayesian Network. In *Proc. of The Young Researchers Track at the 9th International Conference on Intelligent Tutoring Systems*.

Effective Skill Assessment Using Expectation Maximization in a Multi Network Temporal Bayesian Network

Zachary A. Pardos

Advisors: Neil T. Heffernan – Carolina Ruiz – Joseph E. Beck

zpardos@wpi.edu – nth@wpi.edu – ruiz@cs.wpi.edu – joseph.beck@educationaldatamining.org

Abstract. We propose a temporal multi network single skill model for effective assessment and prediction of student skills that is more accurate than multi skill conjunctive models while requiring only a fraction of the computational resources to run. Using the Expectation Maximization algorithm we define steps for effective learning of model parameters over 150,000 responses of real student data that reveal important skill knowledge and learning trends. This skill report is exhibited in the paper. Lastly we focus on how to harness the power of the learned parameters to accurately predict an end of year standardized state math test. Our results of prediction using the multiple single skill network model beat out previous best prediction errors using a single multi skilled model. We believe these results could encourage wider use of these machine learning techniques that can now be effectively run on standard computing machines such as PCs found in school computer labs.

1 Introduction

Drawing powerful inferences from the data is essential to the promise of effective ITS [9]. In the context of Bayesian networks, proper learning and use of learned parameters in the system is key to effective assessment and prediction of student knowledge. This paper will focus on the power of these parameters from two perspectives: assessment and prediction. These perspectives will be visited in the two research questions of the paper. We present results of parameter learning with a multi network Bayesian network from real student data from a web based math tutoring system called ASSISTment [8].

The first research question is "What useful information can we learn from the data?" To answer this question we will use all the student data to learn the parameters of the temporal Bayesian network which is synonymous with a Dynamic Bayesian Net (DBN). A close evaluation of these learned parameters will add insight into the skill background of the 8th grade math students in our dataset. We will also be able to depict learning rates of various skills throughout the year and identify skills that these students as a whole struggle with. An extensive summary of learned parameter findings will be exhibited.

The second research question is "How can we accurately predict end of year state test scores?" This question involves predicting the end of year state math test score for each of the corresponding students in our dataset. This effort will again focus on parameters and empirical results about which combination of learned parameters can be harnessed to achieve the lowest prediction error. For this experiment, the parameters will be learned with a hold out set (half the data) and assessment and prediction will be run with the test set (the other half). Results on numerous predictive models will be presented.

As a platform for our assessment and prediction we propose a temporal Bayesian network topology of separate single skill networks that reduce the required computational resources exponentially. We will also describe data pre-processing methods that allow us to retain all the power of the more complex multi skill model in our proposed single skill models.

1.1 Introduction to the ASSISTment math tutor system

The ASSISTment system is an e-learning and e-assessing system [4] that provides web based math tutoring to 8th-10th grade students. In the 2004-2005 school year, 600 students used the system about once every two weeks. Eight math teachers from two schools would bring their students to the computer lab, at which time students would be presented with randomly selected math questions based on previous years' state test items. In 2004 the system was relatively new and content was still being developed, much of it based on the types of questions that teachers requested. Each tutoring item, which we call an ASSISTment, is based upon a publicly released state test item which we have added tutoring to in the form of sub questions called scaffolding and hints. If students get the item correct they are advanced to the next question. If they answer incorrectly, they are provided with a small tutoring session where they are asked to answer a few questions that break the problem down into steps. The first scaffolding question appears only if the student gets the item wrong. Most test questions that were presented as multiple-choice were converted into text-input questions to reduce the chance of guess. As a matter of logging, the student is only marked as getting the item correct if they answer the question correctly on the first attempt.

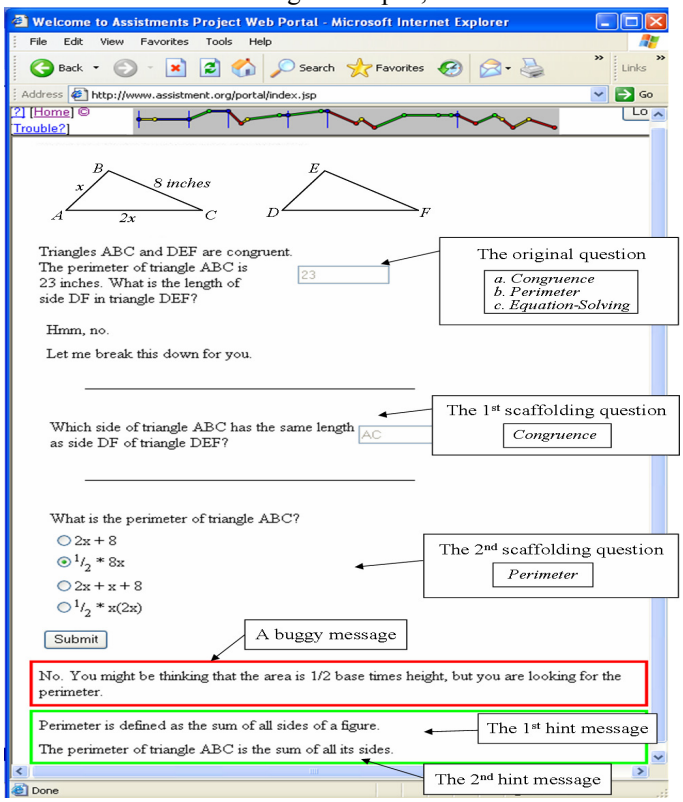

Fig. 1. An example problem in the tutoring system

Subject matter experts in the ASSISTment project developed a set of 106 skills [7] to represent the 8th grade math content that was seen in previous state tests. Each original and scaffold question has been tagged with one or more of these skills and the tagging is part of our dataset described in the following section.

1.2 About our dataset

Our dataset is from logged student use of the ASSISTment tutoring system during the school year 2004-2005. We have around 150,000 question responses from 600 students who went to their school's lab once or twice per month to use the web based tutor system. Figure 2 shows the attributes we have for each of the question responses and an example of what the response data would look like for the two triangles problem in Figure 1.

Table 1. Sample of the question response dataset

UserID	QuestionID	Skill Tags	Correct on first attempt	Timestamp	Is scaffold
1234	3693	Equation-solving, Congruence, Perimeter	0	13-OCT-04 12:20.53	0
1234	3694	Congruence	1	13-OCT-04 12:21.20	1

1234	3695	Perimeter	0	13-OCT-04 12:23.57	1

The other dataset we have, that will be used for research question #2, is the end of year state math test scores for the 600 students who used the system. This test is known as the Massachusetts Comprehensive Assessment System or MCAS. The dataset contains each student's item level response data from the test and the format of that data is shown in Figure 3. Eight of the 600 students were absent from one of the two test sessions and thus data for these students has been pruned leaving us with 592 students.

Table 2. Sample of the state test response dataset

UserID	QuestionID	Skill Tags	Type	Is correct
1234	2005-19	Symbolization Articulation	MC	1
1234	2005-20	Evaluating Functions	MC	1
1234	2005-21	Stem and Leaf Plot	MC	0

While there are some essay and short answer questions on the test, we will only attempt to predict the 29 questions of type multiple choice (MC). The items on the test were tagged with skills after the items had been publicly released by the state. Both the test items and tutor items were tagged with skills from the set of 106, however, the number of skills that have data points in our dataset is only 76.

2 The Temporal Bayesian Model

So why go temporal? The ability to model learning is a big reason. Accounting for time is the only way to be able to track or predict learning over the course of the school year. It also allows us to give more weight to the most recent student responses which can be very valuable in making accurate skill assessments for end of the year predictions. Our previous efforts have focused on doing assessment with a single static network that contains all the skills and questions of the system [4]. The static network consisted of 106 skills and over 1,400 question nodes that could have numerous skills as parents. Learning parameters temporally in a network this size we found to be computationally intractable (the memory requirements were many times greater than the high performance computing machines we had access to). This gave us an opportunity to explore other designs for the temporal network. The goals for the new design were to be far less resource intensive while retaining as much of the inference power of the original design as possible. The solution we arrived at satisfied both goals by splitting the single temporal network into 106 independent skill networks. The details of this design are described below.

2.1 Representing a multi skill network with many single skill networks

Achieving a viable Bayesian network framework with temporality meant departing from our large and multi tagged network of 106 skill nodes and roughly 1,400 question and gate nodes. To do this we split the single network into 106 independent networks; one for each skill. All questions relating to the skill of a network were also placed into that network; multi tagged questions included. Notice that the same "Original Question" in Figure 2 now appears in each of the three independent networks as shown in Figure 3. We no longer use AND gates because there is only a single skill in each network.

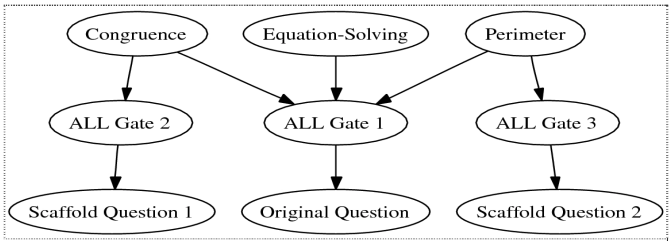

Fig. 2 Example of the original multi skill single network design

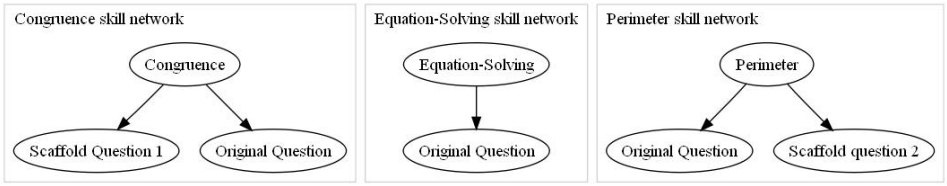

Fig. 3 Three separate single skill networks that represent the multi skill design

So what power is lost or gained in going to single skill networks? To start, the inference power that is kept intact is the ability of the networks to credit or blame a skill depending on if a student gets a singly tagged question correct or incorrect. This inference should remain the same. Another case that is also accounted for in the single skill networks is assigning credit to multiple skills if a student answers a question correctly that was multi tagged in the multi skill network. In this case, the skills relating to the question will be credited in each of the independent networks. The AND gate asserted the same credit to all skills in the multi skill network when a multi tagged question was answered correctly. The case that represents a loss of inference power in the single skill networks is when a question that was multi tagged is answered incorrectly. In this case the single skill networks have no way to selectively assign blame to certain skills. For example it could be that a lack of knowledge of only one of the skills is to blame for the incorrect response. In the multi skill network, if the original question in Figure 2 was answered incorrectly but the two scaffold questions were answered correctly, the Bayes net would assign a higher proportion of blame to the skill of Equation-Solving since the skills of Congruence and Perimeter each have evidence of knowledge supporting them. We have reason to believe that this case can be completed side stepped without loss of accuracy, described below.

2.2 Preserving the inference power from the multi skill conjunctive model

In previous modeling experiments with the static multi skill network we found that actually ignoring incorrect answers to original questions improved the model's predictive fit, see Table 3.

Table 3. State test prediction results when filtering out different types of student responses in the static multi skill network.

Type of filtering	Error
Filter out incorrect originals	13.30%
Use all data	14.45%
Filter out incorrect scaffolds	14.77%
Filter out all scaffolds	15.03%
Filter out all originals	15.32%

The success of this filtering can be attributed to the design of the ASSISTment tutoring system. If an original question is answered incorrectly, the question is broken down into sub questions which separate the skills involved. This was meant to both tutor the student and identify the skill

or skills the student is struggling with. Ignoring the incorrect original question and letting the scaffold questions assign blame appears to be an effective strategy.

This strategy is highly relevant to our single skill network design. Since only original questions are multi tagged, ignoring all incorrect answers to original questions effectively eliminates the one case of blame assignment that the single skill networks could not gracefully handle. We validated this strategy as remaining superior to using all data for state test prediction with the single skill networks.

2.3 Network topology

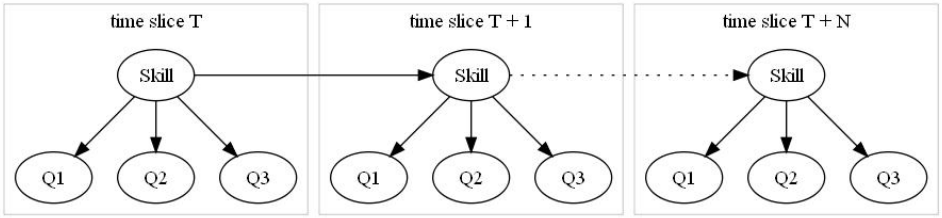

Fig. 4 An example of one of 106 skill temporal networks in the model

As described above, the 106 skills in the model are split into independent Bayesian networks. Figure 4 shows the Hidden Markov Model style of topology of an example network. Only the topology of the first slice is defined, subsequent slices will be copies of the first slice with the exception that the conditional probability table (CPT) of the skill in slice T will be different from the CPT of skills in slices T + 1 to N. This is because the skill in the first time slice has no parent and thus has only a single parameter specifying prior knowledge while skills in subsequent slices all have parents. The CPTs for every subsequent skill consist of the parameters defining the probability of learning and forgetting between each time slice. This temporal parameter is constant between each time slice and because of this it defines a linear function of learning. The CPTs for the question guess/slip parameters are also constant in each time slice. The number of time slices does not need to be specified in advance, it is dynamic and will change with respect to the input.

2.4 The Parameters / Conditional Probability Tables (CPTs)

The second part of specifying a temporal Bayes net is the parameters. Before we give them a value or learn them we need to specify what the parameters in the network are. There are three: prior knowledge of the skill, guess/slip of the questions and the temporal CPT for the skill.

The prior knowledge parameter is a single number that represents the probability that the skill was known previous to the observations available to the network. We will learn this parameter to get an idea of the level of incoming skills among the 8^{th} grade math classes that used our system.

The second parameter is the guess and slip CPT for the question. This table specifies two parameters. One is the guess parameter which is the probability that a question will be answered correctly if the student is believed to not know the skill involved. The second parameter in this CPT is the slip parameter which is the probability that a question will be answered incorrectly if the student is believed to know the skill involved. This table has the largest impact on posterior inferences for predicting student answers to questions and also affects how the network weighs answers to questions when inferring the knowledge of a student. It is possible to have a separate guess and slip CPT for each question in the network but without enough data, the blow up in the number of parameters would decrease accuracy and reliability of the learned parameters. We chose to have a single CPT that represents all questions in a skill network. It is relevant to note that from a modeling perspective we will in effect be learning a guess and slip value per skill as opposed to a guess and slip value per question.

The last parameter is the temporal CPT. This CPT also specifies two parameters: the learning and forgetting rate. The learning parameter specifies the probability that a skill in time slice T will

be known if it was NOT known in the previous time slice. The unlearning or forgetting parameter specifies the probability that a skill in time slice T will be NOT known if it was known in the previous time slice. It is generally not believed, in the literature, that forgetting is a significant factor in tutoring and thus most cognitive models assume no forgetting (probability zero of forgetting). We do not assert that forgetting is a crucial element however it could be valuable to inform teachers of the skills that appear to slip throughout the course of a year. We did however experiment with turning forgetting on and off and present the results of both in section 4.

2.5 Time Slice Representation

An important detail in the specification of a temporal Bayes net is the time slice and what it represents. It is also a very open ended decision as to what time slice should be. In order to measure learning between items, Corbett and Anderson's work gave each item its own time slice [3]. So a time slice was a single answer to a question. In the context of the ASSISTment system, some complexity would need to be added to the model in order to account for the current circumstance that all students' data is collected over various days, sometimes weeks or even months apart. A time slice for each answer would not by itself account for expected learning that occurs outside of the system between sessions. A time slice that represents some granularity of uniform time would account for external learning. The time slice could correspond to a calendar day or week or month for example.

In our implementation we chose the time slice to be a session at the lab. This is somewhat of a middle ground between per item time slices and uniform time. This representation will tell us the probability that learning occurs between sessions. While we do not make the claim that this is the most accurate means of representing time slices it does provide the type of learning information that teachers are likely to find very useful. By tracking learning between sessions teachers can focus their agenda on a certain skill or try different teaching strategies between sessions and come back to the lab with their students and view the change in learning rate for that skill.

3 Research Question #1: What useful information can we learn from the data?

The amount of data collected for the 2004-2005 school year gives us a tremendous opportunity to learn valuable information from the data that will be useful to teachers, students and to content creators (and principal investigators) of the tutoring system. All the information will be contained in the learned parameters of the skill networks. The learned prior knowledge parameter will give teachers a list of the most and least well known skills of the 2004 incoming 8^{th} grade math students which can help guide their curriculum focus in and outside of the tutor system. The guess and slip parameters will point out which skills have questions that tend to be easy to guess and others which require extra care in answering. Finally the learning and forgetting parameters will inform teachers of the learning rates for various skills so they may track the progress of their class and identify skill areas that are stagnant or even declining. While this analysis is being done post-hoc, the same methodology can be applied to aid teachers with their current year class.

3.1 Parameter learning methodology

After an understanding of the parameters and their significance is gained, the parameter learning process is fairly straightforward. The more significant challenge in this process is organization and preparation of the data. Parameters are learned for each of the 106 skill networks independently. An array of cases is defined that serve as the data for the Expectation Maximization (EM) parameter learning. Each case represents a student and all their answers to questions that exist in the particular skill network being evaluated. Each case is a two dimensional array with columns representing the question nodes in the network and rows representing the sessions the student

spent at lab. A value of 1 for a given question and session denotes a correct answer and 0 an incorrect answer. Since students do not see repeats of questions, there will be only one value per question column at most. The input to the EM routine is the temporal Bayes net topology with parameter starting positions included, the array of data cases and the max iteration count. What the EM routine returns when it converges, or when the max iteration is reached, is the same Bayes net topology that was given except now with the learned parameters filled in instead of the starting position values. Viewing the skills with the highest or lowest learned priors is a matter of collecting and ordering the learned parameters from the 106 skill networks, as was done for the results to follow.

While the Expectation Maximization algorithm is not massively parallel, computing the learned parameters for each of the individual skill networks is. We were able to learn parameters for 20 skills simultaneously and complete the parameter learning of all skills in less than 24 hours, a task that would have taken weeks with the multi skill network.

3.2 Results

Skills with lowest Guess

Circumference	0.0850
Isosceles-Triangle	0.0867
Measurement	0.0872
Meaning-of-PI	0.0880
Area-of-Circle	0.0906

Skills with highest Guess

Exponents	0.2389
Interpreting-Linear-Equations	0.2375
Addition	0.2372
Of-Means-Multiply	0.2253
Triangle-Inequality	0.2004

Skills with lowest Slip

Qualitative-Graph-Interpretation	0.0496
Statistics	0.0499
Reduce-Fraction	0.0500
Simple-Calculation	0.0500
Equilateral-Triangle	0.0503

Skills with highest Slip

Number-Line	0.1801
Probability	0.1724
Square-Root	0.1657
Properties-of-Geometric-Figures	0.1493
Exponents	0.1446

Skills with lowest Priors

Venn-Diagram	0.0011
Pythagorean-theorem	0.0087
Of-Means-Multiply	0.0107
Interpreting-Linear-Equations	0.0116
Fraction-Multiplication	0.0196

Skills with highest Priors

Addition	0.8738
Ordering-Numbers	0.8083
Multiplication	0.6966
Integers	0.6854
Multiplying-Positive-Negative-Numbers	0.6655

Skills with lowest Learning

Rate	0.0125
Sum-of-Interior-Angles-Triangle	0.0147
Equation-Concept	0.0166
Venn-Diagram	0.0189
Unit-Conversion	0.0204

Skills with highest Learning

Multiplication	0.3594
Point-Plotting	0.3024
Addition	0.2752
Square-Root	0.2412
Proportion	0.1843

Skills with lowest Unlearning

Integers	0.0305
Multiplying-Positive-Negative-Numbers	0.0360
Ordering-Fractions	0.0451
Qualitative-Graph-Interpretation	0.0475
Statistics	0.0496

Skills with highest Unlearning

Reading-graph	0.4720
Combinatorics	0.4155
Probability	0.3608
Proportion	0.3583
Sum-of-Interior-Angles-Triangle	0.3516

Skills with lowest (Learning minus Unlearning)

Reading-graph	-0.404
Combinatorics	-0.391
Sum-of-Interior-Angles-Triangle	-0.3369
Area	-0.2901
Rate	-0.2897

Skills with highest (Learning minus Unlearning)

Integers	0.1461
Point-Plotting	0.1322
Multiplying-Positive-Negative-Numbers	0.1119
Multiplication	0.1106
Ordering-Numbers	0.1071

Fig. 5 EM Parameter Learning Report

4 Research Question #2: How can we accurately predict end of year state test scores?

Reporting of general skill assessments is an important feature for teachers but teachers also want to know how individual students are doing in class and how well they are mastering the competencies that are being taught. Since the "No Child Left Behind Act", a substantial amount of class time is now being focused on standardized test preparation and knowing if a student is ready to pass the test is of greater concern to teachers now more than ever. The remainder of the paper is dedicated to optimizing this prediction processes using the same multi network skill models described in previous sections.

4.1 Assessment and prediction methodology

To make a prediction of a student's end of year test score we first need to make an assessment of their skills from use of the tutor system. In order to optimize the assessment the parameters of the 106 skill networks must be learned. We created two equal sets of users from the original 592, set A and set B, that were stratified so that each set has an equal distribution with regard to number of questions answered on the system. We then ran the same EM parameter learning method in section 3 for each of the networks but now only using the data from set A. The next step is to assess the knowledge level of each individual student from set B for each of the 106 skills. This is done by providing, to each skill network, a 2D array of student session and question node pairs. The 106 skill assessments for the student are then compiled. The area where our multi network model encounters potential issues is when answers to multi tagged questions need to be predicted. The state test has multi tagged questions that will need to be predicted by our single skill networks. To overcome this problem we create a single static multi skill network to represent the test then, for every user in set B, we import the student's skill values from the single skill networks as soft, probabilistic evidence into the test network. Inferences are then made about the probability of the student answering each of the questions correctly. The probabilities are then summed to get a total predicted score for each student. The Mean Absolute Difference or MAD is reported in the next section.

4.2 Results

Our initial results showed error rates higher than that of the static network counterpart (see Table 4). Another observation from the results was that the network was consistently under predicting student scores by five full points. What first came to mind was that the choice of time slice representation was not ideal for this prediction method. We looked at individual student assessments and a common problematic circumstance identified was when a student had numerous time slices, typically around 8, but only encountered problems for a given skill during one of the earlier sessions. If a student answers questions pertaining to a given skill in session one then that assessed skill level will gradually change after each consecutive session as dictated by the learning/unlearning parameter for that skill. A student's number of sessions is constant across all of their skill networks. To avoid the occurrence of students' skills changing without evidence we set the probability of forgetting to 0% for the results in Table 5.

Table 4. State test prediction results when filtering out different types of student responses in the static multi skill network.

Parameters Used	Filter Incorrect Originals	Error	MAD	Under/Over Prediction
All except priors set at 0.50	YES	17.44%	5.06	-3.5
All learned parameters	YES	18.01%	5.22	-3.9

All learned parameters	NO	19.77%	5.73	-4.7
All except priors set at 0.50	NO	19.25%	5.58	-4.3

The learned guess and slip parameters are used for all of the predictions made. Filtering out incorrect originals was used for all results in Table 5.

Table 5. State test prediction results when setting the probability of forgetting to 0%

Parameters Used	Error	MAD	Under/Over Prediction
Prior set to 0.50, Learning set to 0%	12.72%	3.69	-0.2
Prior set to learned value, Learning set to learned value	13.93%	4.04	1.4
Prior set to 0.50, Learning set to learned value	16.01%	4.66	3.2

The best performing model was the one where we used only the learned guess and slip and set the probability of learning and forgetting to 0%. The accuracy of this model surpasses our best prediction results with the static multi skill network. Setting learning and forgetting to 0% does not inhibit the network from inferring a higher or lower probability of knowledge in subsequent time slices, it just prevents the network from raising or lowering that assessment without encountering new data for the student.

There could be curiosity over what the results would look like if we rounded the probability of a correct answer per item instead of using the probabilistic number. We reran all predictions and rounded to predict either a 1 (for correct) or 0 (for incorrect). To our delight the error results did not differ by more than five tenths of a percentage point from their non rounded counterpart. The result with the largest difference was the "all parameters with no unlearning" with a 13.88% error when rounding vs. the 13.93% without rounding.

4 Contribution

We demonstrated how meaningful parameter values can be learned from data in a temporal Bayesian network model. We have also shown how those parameters can be used to achieve accurate per student assessment and prediction and how multi skill prediction can be done by rolling up the temporal networks into a single static multi skill network. Finally we proposed a resource efficient single skill multi network model design and data filtering strategy that retains the power of a computationally intractable multi skill network.

It is significant to point out the computing resource benefits to this multiple single skill network design over a single multi skill network. It was not feasible to run the multi skill network temporally on our computing system equipped with 16 processors and 32 gigabytes of memory, however, there are smaller multi skill networks that are able to be run on this system but not the majority of lesser equipped systems. Going to a single skill design not only reduces resources such that the computation is feasible on a strong machine but it reduces the resources exponentially so that the task can be handled by a common personal computer. This opens the doors for real-time application of temporal skill tracking that could be done centrally for many students or offloaded to the student's lab computer at school.

Acknowledgements

We would like to thank Joseph E. Beck, Worcester Public Schools and all of the people associated with creating the ASSISTment system listed at www.ASSISTment.org including investigators Kenneth Koedinger and Brian Junker at Carnegie Mellon. We would also like to acknowledge funding from the US Department of Education, the National Science Foundation, the Office of Naval Research and the Spencer Foundation.

References

1. Anozie N. & Junker, B. (2006, in press). Predicting End-of-year Accountability Assessment Scores from Monthly Student Records in an Online Tutoring System. Workshop on Educational Data Mining held at the 21st National Conference on Artificial Intelligence (AAAI), Boston, 2006.
2. Barnes, T., (2005). Q-matrix Method: Mining Student Response Data for Knowledge. In Beck. J (Eds). *Educational Data Mining: Papers from the 2005 AAAI Workshop.* Technical Report WS-05-02. ISBN 978-1-57735-238-9.
3. Corbett, A. T., Anderson, J. R. & O'Brien, A. T. (1995) Student modeling in the ACT programming tutor. Chapter 2 in Nichols, P. D., Chipman, S. F. and Brennan, R. L. (eds.) (1995). Cognitively diagnostic assessment. Lawrence Erlbaum Associates: Hillsdale, NJ.
4. Feng, M., Heffernan, N. T., Mani, M., & Heffernan, C. (2006). Using Mixed-Effects Modeling to Compare Different Grain-Sized Skill Models. In Beck, J., Aimeur, E., & Barnes, T. (Eds). Educational Data Mining: Papers from the AAAI Workshop. Menlo Park, CA: AAAI Press. pp. 57-66. Technical Report WS-06-05. ISBN 978-1-57735-287-7.
5. Manske, M., & Conati, C.: 2005, Modeling Learning in Educational Games*, AIED'05, 12th International Conference on AI in Education*, Amsterdam.
6. Pardos, Z.A., Heffernan, N.T., Anderson, B., Heffernan, C.L. The Effect of Model Granularity of Student Performance Prediction Using Bayesian Networks. In the Complete On-Line Proceedings of the Workshop on Data Mining for User Modeling, at the 11th International Conference on User Modeling (UM 2007) Pages 91-100
7. Razzaq, L., Heffernan, N., Feng, M., Pardos Z. (2007). Developing Fine-Grained Transfer Models in the ASSISTment System. Journal of Technology, Instruction, Cognition, and Learning, Vol. 5. Number 3. Old City Publishing, Philadelphia, PA. 2007. pp. 289-304.
8. Razzaq, Heffernan, Koedinger, Feng, Nuzzo-Jones, Junker, Macasek, Rasmussen, Turner & Walonoski (2007). Blending Assessment and Instructional Assistance. In Nadia Nedjah, Luiza deMacedo Mourelle, Mario Neto Borges and Nival Nunesde Almeida (Eds). Intelligent Educational Machines within the Intelligent Systems Engineering Book Series. 23-49 Springer Berlin / Heidelberg.
9. VanLehn, K. (2006). The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education, 16*, 227–265