# Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement

**Conference Paper** *in* Lecture Notes in Computer Science · June 2006

DOI: 10.1007/11774303_17 · Source: DBLP



Kenneth R. Koedinger
Carnegie Mellon University


Brian Junker
Carnegie Mellon University

# Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement

Hao Cen, Kenneth Koedinger, Brian Junker

Carnegie Mellon University
5000 Forbes, Pittsburgh, PA, U.S.A.
hcen@andrew.cmu.edu, koedinger@cmu.edu, brian@stat.cmu.edu

**Abstract.** A cognitive model is a set of production rules or skills encoded in intelligent tutors to model how students solve problems. It is usually generated by brainstorming and iterative refinement between subject experts, cognitive scientists and programmers. In this paper we propose a semi-automated method for improving a cognitive model called Learning Factors Analysis that combines a statistical model, human expertise and a combinatorial search. We use this method to evaluate an existing cognitive model and to generate and evaluate alternative models. We present improved cognitive models and make suggestions for improving the intelligent tutor based on those models.

## 1 Introduction

A cognitive model is a set of production rules or skills encoded in intelligent tutors to model how students solve problems. (Production, skill, and rule are used interchangeably in this paper.) Productions embody the knowledge that students are trying to acquire, and allows the tutor to estimate each student's learning of each skill as the student works through the exercises [4].

A good cognitive model captures the fine knowledge components in a curriculum, provides tailored feedback and hints, select problem with difficulty level and learning pace matched to individual students, and eventually, improves student learning. However, initial models are usually generated by brainstorming and iterative refinement between subject experts, cognitive scientists and programmers. These first pass models are best guesses and our experience is that such models can be improved.

In this paper, we propose a method called Learning Factors Analysis (LFA) and use it to answer three questions relevant to the field of intelligent tutoring systems.

1. How can we describe learning behavior in terms of an existing cognitive model? We need to identify the initial difficulty level of each production and how fast can a student learn each rule (i.e., what is the learning rate). We can then provide parameters that indicate student performance on this set of rules and how that performance improves with practice and instruction on those rules.

2. How can we evaluate and improve a cognitive model in an inexpensive way? We need to identify the causes of the deviation from the deterministic cognitive model, define the measures of a model's complexity and fit, and mine the student-tutor log data.

3. How can we use the information from LFA to improve the tutor and the curriculum? We need to identify over-taught or under-taught rules, and even "hidden" knowledge components within them. As a result, we can adjust their contribution to curriculum length without compromising student performance.

## 2 Literature Review

One measure of the performance of a cognitive model is how the data fit the model. Newell and Rosenbloom found a power relationship between the error rate of performance and the amount of practice [13]. Depicted by equation (1), the relationship shows that the error rate decreases according to a power function as the amount of practice increase. The curve for the equation is called a "learning curve".

$$Y = aX^{b}. \quad \textbf{(1)}$$

where
Y = the error rate
X = the number of opportunities to practice a skill
a = the error rate on the first trial, reflecting the intrinsic difficulty of a skill
b = the learning rate, reflecting how easy a skill is to learn

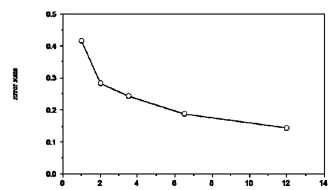

**Fig. 1.** A power law learning curve

The learning curve model has been used to visually identify non-obvious or "hidden" knowledge components. Corbett and Anderson observed that the power relationship might not be readily apparent in some complex skills, which have blips in their learning curves [5], as shown in figure 2. They also found the power relationship holds if the complex skill can be decomposed into subskills, each of which exhibits a smoother learning curve.

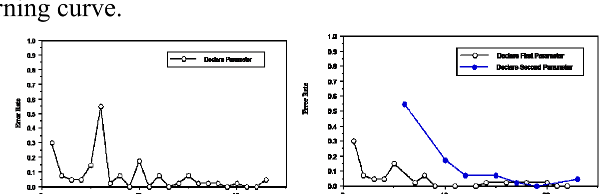


**Fig. 2.** A learning curve with blips (*left*) split into two smoother learning curves (*right*)

As seen in the graphs above, the single production Declare-Parameter produces a learning curve with several blips. However by breaking it into two more specific productions, Declare-First-Parameter and Declare-Second-Parameter, the model becomes more fine-tuned and recognizes that the skills are different. The knowledge decomposition (considering parameter position) that was non-obvious from the original model became revealed on closer inspection of learning curve data.

Other approaches to model refinement include having a simulated student to find incorrect rules and to learn new rules via human tutor intervention [16], using theory refinement to introduce errors to models incorrect student behaviors [1] and using Q-matrix to discover knowledge structure from student response data [15,2]. Compared with the simulated student approach, our method does not require building a simulated student. The theory refinement approach starts with an initial knowledge base and keeps correcting errors in the knowledge base from error examples until the knowledge base is consistent with the examples. It may lead to overfit the examples. The Q-matrix approach was used to automatically extract features in the problem set. The model found by this approach may be similar to the model adding or merging difficulty factors in our method.

## 3 The Cognitive Model and its Data under Investigation

We illustrate the LFA methodology using data obtained from the Area Unit of the Geometry Cognitive Tutor (see http://www.carnegielearning.com). The initial cognitive model implemented in the Tutor had 15 skills that correspond to productions or, in some cases, groups of productions. The productions are

Circle-area – Given the radius , find the area of a circle

Circle-circumference – Given the diameter, find the circumference of a circle.

Circle-diameter -- Given the radius or circumference, find the diameter of a circle.

Circle-radius -- Find the radius given the area, circumference, or diameter.

Compose-by-addition – In a+b=c, given any two of a, b, or c, find the third.

Compose-by-multiplication – In a*b=c, given any two of a, b, or c, find the third.

Parallelogram-area – Given the base and height, find the area of a parallelogram.

Parallelogram-side – Given the area and height (or base), find the base (or height).

Pentagon-area – Given a side and the apothem, find the area of a pentagon.

Pentagon-side – Given area and apothem, find the side (or apothem).

Trapezoid-area – Given the height and both bases, find the area of a trapezoid.

Trapezoid-base – Given area and height, find the base of a trapezoid.

Trapezoid-height – Given the area and the base, find the height of a trapezoid.

Triangle-area – Given the base and height, find the area of a triangle.

Triangle-side – Given the base and side, find the height of a triangle.

Our data consist of 4102 data points involving 24 students, and 115 problem steps. Each data point is a correct or incorrect student action corresponding to a single production execution. Table 1 displays typical student action records in this data set. It has five columns – student, success, step, skill, and opportunities. Student is the names of the students. Success is whether the student did that step correctly or not in the fist attempt. 1 means success and 0, failure. Step is the particular step in a tutor problem the students are involved in. “p1s1” stands for problem 1 step 1. Skill is the production rule used in that step. Opportunities mean the number of previous times to use a particular skill. It increments every time the skill is used by the same student, and can be computed from the first and fourth columns.

**Table 1.** The sample data

| Student | Success | Step | Skill | Opportunities |
|---|---|---|---|---|
| A | 0 | p1s1 | Circle-area | 1 |
| A | 1 | p2s1 | Circle-area | 2 |
| A | 1 | p3s1 | Circle-area | 3 |

## 4. Learning Factor Analysis

LFA has three components: a statistical model that quantifies the skills, the difficulty factors that may affect student performance in the tutor curriculum, and a combinatorial search that does model selection.

### 4.1 The Statistical Model

The power law model applies to individual skills and does not typically include student effects. Because the existing cognitive model has multiple rules, and the data contains multiple students, we made four assumptions about student learning to extend the power law model.

1. Different students may initially know more or less. Thus, we use an *intercept* parameter for each student.

2. Students learn at the same rate. Thus, *slope* parameters do not depend on student. This is a simplifying assumption to reduce the number of parameters in equation 2. We chose this simplification, following Draney, Wilson and Pirolli [7], because we are focused on refining the cognitive model rather than evaluating student knowledge growth.

3. Some productions are more likely to be known than others. Thus, we use a *intercept* parameter for each production

4. Some productions are easier to learn than others. Thus, we need a *slope* parameter for each production

Based on the assumptions, we developed a multiple logistic regression model.

$$\ln\left(\frac{p}{1-p}\right) = \sum \alpha_i X_i + \sum \beta_j Y_j + \sum \gamma_j Y_j T_j \,. \qquad \textbf{(2)}$$

Where

p = the probability to get an item right

what does covariates mean here?

X = the covariates for students

Y = the covariates for skills

T = the covariates for the number of opportunities practiced on the skills

Y T = the covariates for interaction between skill and the number of practice opportunities for that skill

α = the coefficient for each student, i.e. the student intercept

β = the coefficient for each rule, i.e. the production intercept

γ = the coefficient for the interaction between a production and its opportunities, i.e. the production slope

### 4.2 Difficulty Factors

A difficulty factor refers specifically to a property of the problem that causes student difficulties (e.g., first vs. second parameter in figure 3). By assessing the performance difference on pairs of problems that vary by one factor at a time, we can identify the hidden knowledge component(s) that can be used to improve a cognitive model [9]. Difficulty factors have been used to empirically evaluate a small number of alternative models [6, 10, 11].

In our study, subject experts identified four multi-valued factors for the Area Unit of the Geometry Tutor. Table 2 lists their names and values.

**Table 2.** Factors for the Area Unit and their values

| Factor Names | Factor Values |
|---|---|
| Embed | alone, embed |
| Backward | forward, backward |
| Repeat | initial, repeat |
| FigurePart | area, area-difference, area-combination, diameter, circumference, radius, side, segment, base, height, apothem |

"Embed" indicates whether a shape is embedded in another shape. Consider two tutor problems requiring the same production rule CIRCLE-AREA at some step in the problem. In one of the problems, the circle is embedded in a square; while in the other one, the circle is presented alone. Students may find it harder to find the area of circle when it is embedded in another figure because extra effort is necessary to find the circle and its radius. "Backward" means whether the production rule to be used is in its backward form of a taught formula, or its forward form. The forward form of Compose-by-addition is S = S1 + S2, and its backward forma is S1 = S - S2. "Repeat" indicates whether the production rule has been used previously in the same problem. "FigurePart" indicates the part of the figure in the geometry shape to be computed.

### 4.3 Combinatorial Search

The goal of the combinatorial search is to do model selection within the logistic regression model space [8]. Difficulty factors are incorporated into an existing cognitive model through a model operator called Binary Split, which splits a skill a skill with a factor value, and a skill without the factor value. For example, splitting production Circle-area by factor Embed with value alone leads to two productions: Circle-area with the factor value alone (called Circle-area*alone), and Circle-area with the factor value embed (Circle-area*embed). Table 3 shows the data before and after a split with Embed.

**Table 3**. The data before and after split. Factors are incorporated in column Skill (after split). The opportunities (after split) change accordingly.

| Student | Step | Skill | OPT | Factor | Skill (after split) | OPT |
|---|---|---|---|---|---|---|
| A | p1s1 | Circle area | 1 | alone | CA-alone | 1 |
| A | p2s1 | Circle area | 2 | embed | CA-enbed | 1 |
| A | P3s1 | Circle area | 3 | alone | CA-alone | 2 |

A* search is the combinatorial search algorithm [14] in LFA. It starts from an initial node, iteratively creates new adjoining nodes, explores them to reach a goal node. To limit the search space, it employs a heuristic to rank each node and visits the nodes in order of this heuristic estimate.

In our study, the initial node is the existing cognitive model. Its adjoining nodes are the new models created by splitting the model on the difficulty factors. We do not specify a model to be the goal state because the structure of the best model is unknown. We do specify the stopping criterion by setting the upper bound of the number of node expansions, for this paper to 50 node expansions per search.

The heuristic guiding the search is one of the two scoring functions for regression models – AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) Each search is run twice, guided by a different heuristic each time. A good model captures sufficient variation in the data but is not overly complicated by balancing between model fit and complexity minimizing prediction risk [17]. AIC and BIC are two estimators for prediction risk, and hence used as heuristics in the search.

$$\text{AIC} = -2*\text{log-likelihood} + 2*\text{number of parameters}. \quad \textbf{(3)}$$

$$\text{BIC} = -2*\text{log-likelihood} + \text{number of parameters} * \text{number of observations}. \quad \textbf{(4)}$$

where log-likelihood measures the fit, and the number of parameters, which is the number of covariates in equation 2**,** measures the complexity. Based on these two formulas, the lower the AIC or BIC, the better the balance between model fit and complexity. BIC puts a more severe penalty for complexity, leading to a smaller model than other methods.

A more interpretable metric for fit is Mean Absolute Deviance (MAD) -- the average of the absolute values of the differences between observed values and predicted values. We do not use it as a heuristic because it leads to over fitting. We include it as a measure of the improvement in the model fit.

Figure 4 illustrates A* search with AIC as the heuristic. The original model is evaluated and AIC is computed. The model is then split into a few new models by incorporating the factors. AICs are computed from each of the new model. A* selects the best one (the shaded node with value 5301) for the next model generation. A* does not always go down. It may go up to select a model (the shaded node with value 5312) to expand if all the new models have worse heuristic scores than a previous model has. After several expansions, it finds a best node with the lowest AIC value.

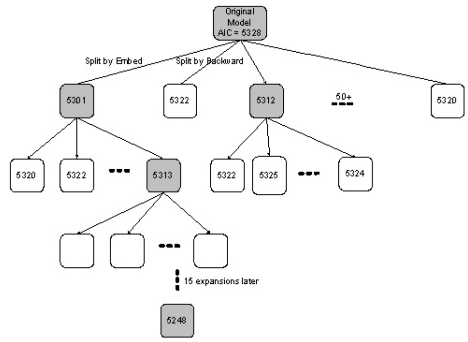


**Fig. 3.** Using A* algorithm search through the model space

## 5. Experiments and Results

### 5.1 Experiment 1

This experiment addresses the question -- How can we describe learning behavior in terms of an existing cognitive model? Specifically, we want to find out the learning rate and initial difficulty level of each rule, and the initial performance of students, given the data. The question is answered by fitting the logistic regression model in equation 2 and getting the coefficients. The coefficient estimates for the skills and students, and the overall model statistics are summarized in table 4.

**Table 4.** Statistics for a partial list of the skills, students and the overall model. Intercept for skill is the initial difficulty level for each skill. Slope is the learning rate. Avg Practice Opportunties is the average amount of practice per skill across all students. Initial Probabltiy is the estimated probability of getting a problem correct in the first opportunity to use a skill accross all students. Avg Probability and Final Probability are the success probability to use a skill at the average amount of opportunities and the last opportunity, respectively.

| Skill | Intercept | Slope | Avg Opportunties | Initial Probability | Avg Probability | Final Probability |
|---|---|---|---|---|---|---|
| Parallelogram-area | 2.14 | -0.01 | 14.9 | 0.95 | 0.94 | 0.93 |
| Pentagon-area | -2.16 | 0.45 | 4.3 | 0.2 | 0.63 | 0.84 |

| Student | Intercept |
|---|---|
| student0 | 1.18 |
| student1 | 0.82 |
| student2 | 0.21 |

| Model Statistics | |
|---|---|
| AIC | 3,950 |
| BIC | 4,285 |
| MAD | 0.083 |

The higher the intercept of the each skill, the lower the initial difficulty the skill has. The higher the slope of the each skill, the faster students learned the skill. Pentagon-area is the hardest skill with the intercept of -2.16. Parallelogram-area is the easiest skill with the intercept of 2.14. Three skills have small slopes close to zero --

Compose-by-addition (-.04) and Parallelogram-area (-.01), Triangle-area (.03). Parallelogram-area was already mastered with an initial success probability .95. It appears that more practice on those skills does not lead to much learning gain. Interestingly, although PENTAGON-AREA is the hardest skill among all, it has the highest learning rate .45, leading to bigger improvement with more practice.

The coefficients for students measure each student's overall performance. The higher the number, the better the student performed. The AIC, BIC and MAD statistics provide a baseline for evaluating alternative models discussed below.

### 5.2 Experiment 2

This experiment addresses the question -- How can we improve a cognitive model? The question is answered by running LFA on the data including the factors, and searching through the model space. The improved models by LFA with BIC are summarized in table 5. The improved models by LFA with AIC is summarized in the interpretation.

**Table 5.** Top three improved models found by LFA with BIC as the heuristic. The table shows the history of splits and model statistics.

| Model 1 | Model 2 | Model 3 |
|---|---|---|
| Number of Splits:3 | Number of Splits:3 | Number of Splits:2 |
| 1. Binary split compose-by-multiplication by figurepart segment<br>2. Binary split circle-radius by repeat repeat<br>3. Binary split compose-by-addition by backward backward | 1. Binary split compose-by-multiplication by figurepart segment<br>2. Binary split circle-radius by repeat repeat<br>3. Binary split compose-by-addition by figurepart area-difference | 1. Binary split compose-by-multiplication by figurepart segment<br>2. Binary split circle-radius by repeat repeat |
| Number of Skills: 18 | Number of Skills: 18 | Number of Skills: 17 |
| AIC: 3,888.67<br>BIC: 4,248.86<br>MAD: 0.071 | AIC: 3,888.67<br>BIC: 4,248.86<br>MAD: 0.071 | AIC: 3,897.20<br>BIC: 4,251.07<br>MAD: 0.075 |

LFA suggests better models, which make finer distinctions on some skills in the original model and identify which difficulty factors the subject experts thought would turn out to be psychologically important. All the better models found by AIC and BIC have better (i.e. lower) statistical scores than those of the original. For the best BIC model, its BIC is reduced by 37, and AIC by 62. The fit of the new model, as measured by MAD, is reduced by .012. The best AIC model reduces AIC by an even larger amount of 83, and increases BIC by 18. Its MAD is reduced by .02.

The improved skills common to most of the better models are Compose-by-multiplication, Compose-by-addition, Circle-area, and Triangle-area. We will discuss a few examples here.

All the new models suggest splitting Compose-by-multiplication into two skills – Cmarea and CMsegment, making a distinction of the geometric quantity being multiplied. By examining the positions of these problems in the curriculum, CMarea at the 43rd step and CMsegement at the 90th. As seen in table 6, although the final probability of CMarea is high .96, the initial probability of CMsegment is low .32. This sudden drop in the success probability at later steps corresponds to a significant blip in the learning curve as illustrated in figure 2. The distinction between different geometric quantities suggests treating the original skill differently. LFA successfully identified the blip without the need of visually inspecting learning curves.

**Table 6.** Success probabilities of CMarea and CMsegment

| | Initial Probability | Avg Probability | Final Probability |
|---|---|---|---|
| CM*area-combination | .64 | .89 | .96 |
| CM*segment | .32 | .54 | .60 |

The subject experts thought embedding a shape into another shape would increase the difficulty of a skill and identified a factor "Embed", hoping LFA could make a distinction on it. LFA split these two skills by Embed in all the top AIC models. The three probabilities of CAalone and CAembed are shown in table 7. Does Embed make find the circle area harder? Note that problems with CAembed are introduced later in the curriculum after students have had significant practice with CAalone, about the time CAalone has reached the average probability of .81. At this point, CAembed has an initial probability of .71, indicating an increase in difficulty.

**Table 7.** Success probabilities of CAalone and CAembed

| | Initial Probability | Avg Probability | Final Probability |
|---|---|---|---|
| CA*alone | .42 | .81 | .93 |
| CA*embed | .71 | .89 | .92 |

### 5.3 Experiment 3

In experiment 2, LFA improved the original model by splitting skills. Experiment 3 addresses model improvement even further -- Will some skills be better merged than if they are separate skills? Can LFA recover some elements of truth if we search from a merged model, given difficulty factors?

We merged some skills in the original model to remove some of the distinctions, which are represented as the difficulty factors. Circle-area and Circle-radius are merged into one skill Circle; Circle-circumference and Circle-diameter into Circle-CD; Parallelogram-area and Parallelogram-side into Parallelogram; Pentagon-area, and Pentagon-side into Pentagon; Trapezoid-area, Trapezoid-base, Trapezoid-height into Trapezoid. The new merged model has 8 skills -- Circle, Crcle-CD, Compose-by-addition, Compose-by-multiplication, Parallelogram, Pentagon, Trapezoid, Triangle.

Then we substituted the original skill names with the new skill name in the data, ran LFA including the factors, and had the A* algorithm search through the model space. The improved models by LFA with BIC are summarized in table 8. The improved models by LFA with AIC are summarized in the interpretation.

**Table 8.** Top three improved models found by LFA with BIC as the heuristic.

| Model 1 | Model 2 | Model 3 |
|---|---|---|
| Number of Splits: 4 | Number of Splits: 3 | Number of Splits: 4 |
| Number of skills: 12 | Number of skills: 11 | Number of skills: 12 |
| Circle *area<br>Circle *radius*initial<br>Circle *radius*repeat<br>Compose-by-addition<br>Compose-by-addition*area-difference<br>Compose-by-multiplication*area-combination<br>Compose-by-multiplication*segment | All skills are the same as those in model 1 except that<br>1. Circle is split into Circle *backward*initial, Circle *backward*repeat, Circle*forward,<br>2. Compose-by-addition is not split | All skills are the same as those in model 1 except that<br>1. Circle is split into Circle *backward*initial, Circle *backward*repeat, Circle *forward,<br>2. Compose-by-addition is split into Compose-by-addition and Compose-by-addition*segment |
| AIC: 3,884.95<br>BIC: 4,169.315<br>MAD: 0.075 | AIC: 3,893.477<br>BIC: 4,171.523<br>MAD: 0.079 | AIC: 3,887.42<br>BIC: 4,171.786<br>MAD: 0.077 |

LFA fully recovered three skills (Circle, Parallelogram, Triangle), suggesting the distinctions made in the original model are necessary. LFA partially recovered two skills (Triangle, Trapezoid), suggesting the some original distinctions are necessary and some are not. LFA did not recover one skill (Circle-CD), suggesting that the original distinctions might not be necessary. LFA recovered one skill (Pentagon) in a different way, suggesting the original distinction may not be as significant as the distinction caused by another factor. We discuss a few examples here.

In BIC model 1, Circle is split into Circle*area, and Circle*radius. The other two BIC models and all the AIC models split it into Circle*backward, and Circle*forward, which are equivalent to Circle-AR*area, and Circle-AR*radius because of the one-to-one relationship between forward and area and between backward and radius. Thus, LFA fully recovers the Circle skills.

None of the models recovered Circle-CD. This suggests that it may not be necessary to have two separate skills for Circle-circumference and Circle-Diameter. It appears that once students learn the formula circumference = $\pi$*diameter, they can fairly easily apply it in the forward or backward direction.

In one of the top AIC models, Pentagon is split into Pentagon*initial and Pentagon*repeat, instead of Pentagon*area and Pentagon*side. This suggests that the distinction between the first use of a Pentagon skill in a problem and later uses of that skill in the same problem may be more significant than the distinction between the area and the side. Usually repeated use of a skill in the same problem is easier than the original use. For instance, once a student makes the *initial* relatively difficult determination that the Pentagon formula is relevant to a problem and recalls it, he need only use it again and perform easier arithmetic in *repeated* opportunities in that same problem.

### 5.4 Combining the results from experiment 1, 2, 3

By combining the results from the three experiments, we can address question 3 -- How can we use LFA to improve the tutor and the curriculum by identifying over-taught or under-taught rules, and adjusting their contribution to curriculum length without compromising student performance?

Parallelogram-side has a high intercept (2.06) and a low slope (-.01). Its initial success probability is .94 and the average number of practices per student is 14.9. Much practice spent on an easy skill is not a good use of student time. Reducing the amount of practice for this skill should save student time without compromising their performance. Trapezoid-height has a low intercept (-1.55), and a positive slope (.27). Its initial success probability is .29 and the average number of practices per student is 4.2. The final success probability is .69, far away from the level of mastery. More practice on this skill is needed for students to reach mastery.

The advantage of LFA goes even further. An original rule may have two split rules, each of which need decidedly different amounts of practice, because they have different initial difficulty and learning rates. However, students who have appeared to master the original rule in the curriculum before even reading the second split rule might not get enough practice on the second split rule. Compose-by-multiplication is such a case, as seen in table 9.

**Table 9.** Statistics of Compose-by- Multiplication before and after split

| | Intercept | slope | Avg Practice Opportunties | Initial Probability | Avg Probability | Final Probability |
|---|---|---|---|---|---|---|
| CM | -.15 | .1 | 10.2 | .65 | .84 | .92 |
| CMarea | -.009 | .17 | 9 | .64 | .86 | .96 |
| CMsegment | -1.42 | .48 | 1.9 | .32 | .54 | .60 |

With final probability .92 students seem to have mastered Compose-by-multimplication. However, the decomposition of the skill shows a different picture. CMarea does well with final probability .96. But CMsegment has final probability only .60 and an average amount of practice less than 2. The knowledge-tracing algorithm in the tutor may let the student go after he reaches the mastery on Compose-by-addition in the original model. But with the model found by LFA, the knowledge-tracing algorithm will be able to catch the weakness of students in acquiring CMsegment.

## 5 Conclusions and Future Work

Learning Factors Analysis is a way to combine statistics, human expertise and combinatorial search to evaluate and improve a cognitive model. The system we have developed is implemented in Java and is able to evaluate a model in seconds and conduct a search evaluating hundreds of models in 4-5 hours. The statistics for each model are meaningful, and the new improved models have better statistical scores and are interpretable. We are planning to use the method for datasets from other tutors to discover its potential for model and tutor improvement.

## 6 Acknowledgements

This research is sponsored by a National Science Foundation grant to the Pittsburgh Science of Learning Center. We thank Joseph Beck, Albert Colbert, and Ruth Wylie for their comments on earlier versions of this paper.

## References

1. Baffes P., Mooney R J.: A Novel Application of Theory Refinement to Student Modeling. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96). Portland OR (1996) 403-408
2. Barnes T.: The Q-matrix Method: Mining Student Response Data for Knowledge. In: Proceedings of AAAI 2005 Educational Data Mining Workshop. (2005)
3. Cen, H, Koedinger, K, Junker B.: Automating Cognitive Model Improvement by A*Search and Logistic Regression. In: Proceedings of AAAI 2005 Educational Data Mining Workshop. (2005)
4. Corbett A.T., Anderson, J.R., O'Brien A.T.: Student Modelling in the ACT Programming Tutor. In: Cognitively Diagnostic Assessment. Erlbaum, Hillsdale, NJ (1995)
5. Corbett, A.T. and Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4. (1995) 253-278.
6. Croteau E.A., Heffernan N. T., Koedinger K.R.: Why are Algebra Word Problem Difficult? Using Tutoring Log Files and the Power Law of Learning to Select the Best Fitting Cognitive Model. In: Proceedings of Intelligent Tutoring Systems. (2004)
7. Draney, K., Pirolli, P., & Wilson, M.: A Measurement Model for a Complex Cognitive Skill. In: Cognitively Diagnostic Assessment. Erlbaum, Hillsdale, NJ (1995)
8. Junker B.W., Koedinger, K., Trottini M.: Finding Improvements in Student Models for Intelligent Tutoring Systems via Variable Selection for a Linear Logistic Test Model., presented at annual meeting of Psychometric Society Vancouver BC (2000)
9. Koedinger K. R.: http://pact.cs.cmu.edu/koedinger/koedingerResearch.html. (2000)
10. Koedinger K. R., Mathan S.: Distinguishing Qualitatively Different Kinds of Leaning Using Log Files and Learning Curves. In: Proc. of Intelligent Tutoring Systems (2004)
11. Koedinger, K. R., Nathan, M. J.: The Real Story Behind Story Problems: Effects Of Representations On Quantitative Reasoning. The Journal of the Learning Sciences, 13 (2). (2004) 129-164.
12. Koedinger K.R., Anderson, J.R., Hadley W. H., Mark M. A.: Intelligent Tutoring Goes to School in the Big City. In: Proceedings of the 7th World Conference on Art, Intelligence and Education, AACE. (1995)
13. Newell, A., Rosenbloom, P.: Mechanisms of Skill Acquisition and the Law of Practice. In Anderson J. (ed.): Cognitive Skills and Their Acquisition, Erlbaum Hillsdale NJ (1981)
14. Russell S., Norvig P., Artificial Intelligence, 2nd edn. Prentice Hall (2003)
15. Tatsuoka, K.: Rule space: An approach for dealing with misconceptions based on item response theory. Journal of Educational Measurement. 20(4) (1983) 345-354.
16. Ur, S., VanLehn, K.: STEPS: A Simulated, Tutorable Physics Student. Journal of Artificial Intelligence in Education. 6(4) (1995) 405-437
17. Wasserman, L.: All of Statistics: A Concise Course in Statistical Inference. Springer (2004)