Automating Cognitive Model Improvement by A*Search and Logistic Regression

Hao Cen, Kenneth Koedinger, Brian Junker
Carnegie Mellon University
5000 Forbes, Pittsburgh, PA, U.S.A.

Abstract

A good cognitive model is important to the effectiveness to an intelligent tutor. In this paper we present a method of combining the A* search algorithm and logistic regression to automate the improvement of a cognitive model by 1) automatically generating different models by mutating learning factors in a base model 2) integrating logistical regression to evaluate different models 3)selecting the best model through a depth-first search algorithm.

1. Introduction

A cognitive model is a set of production rules or skills encoded in intelligent tutors to model how students solve problems. Production rules and skill are used interchangeably in this paper. Production rules 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 (Corbett, Anderson, O'Brien 1992). The model is usually generated by brainstorming and iterative refinement between subject experts, cognitive scientists and programmers. However, these first pass models are best guesses and our experience is that such models can be improved.. In this paper, we present a data-driven approach to evaluate the initial model and to automatically improve it by mining log data of student-tutor interaction. We first introduce related work on cognitive model evaluation, then describe the cognitive model we have analyzed, the methodology we explored, and lessons learned on how best to apply data mining approaches to the problem of cognitive model improvement.

2. Literature Review

One measure of the performance of a cognitive model is how the data fits the model. Newell and Rosenbloom (1993) found the inverse relationship between the error rate of performance and the number of practice -- the error rate decreases as the amount of practice increases. The relationship can be depicted as a power function

$Y = a X^b$
Y – the error rate
X – the opportunity to practice a skill
a – the error rate on the first trial
b – the learning rate

Figure 1 shows a steadily declining learning curve with the x-axis as the opportunity to practice a skill and the y-axis as the error rate.

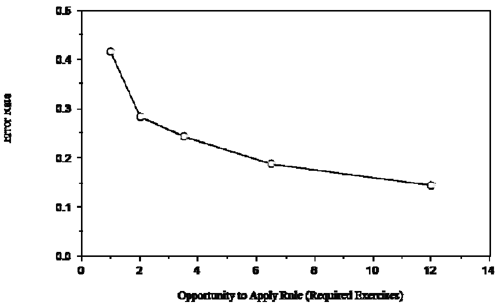

Figure 1 Power Law Learning Curve

Corbett, Anderson, and O'Brien (1992) observed that the power relationship might not be readily apparent in some complex skills, which have blips in their learning curves, as is shown in figure 2. They also found the power relationship holds if the complex skill can be decomposed into some subskills, each of which has a smoother learning curve in figure 3.

In other words, the original model was reasonable for many production rules, but the one shown in Figure 2 (Declare-Parameter) was too general. By breaking the Declare-Parameter production into two more specific productions, Declare-First-Parameter and Declare-Second-Parameter, allows the cognitive model to make a needed distinction (and thus provide better hint messages and do more accurate student modeling).

Koedinger (2000) suggested am empirical method for improving cognitive models that involves experimental comparisons of student error rates on systematic variation of a core problem when just one problem feature or "difficulty factor" (e.g., first vs. second parameter or concrete vs. abstract presentation). He calls this approach "Difficulty Factor Assessment". From theory and task analysis, researchers can hypothesize the likely factors that cause student difficulties. By assessing the performance difference on pairs of problems that vary by one factor, we can identify the hidden knowledge component that can be used to improve a cognitive model.

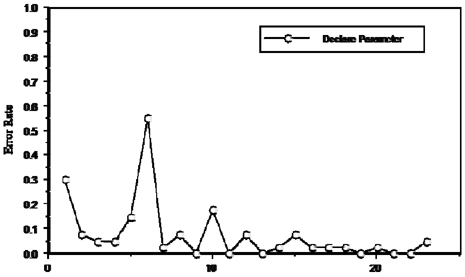

Figure 2 Before Split

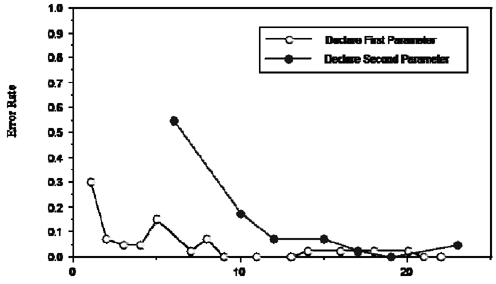

Figure 3 After Split

Inspecting learning curves, like those in Figures 2 and 3, provides an alternative way to identify factors that characterize problem difficulty and, further, characterize how much practice on one problem in the tutor transfers to the next. By considering changes in student performance over time (the "Opportunity" variable on the x-axis in Figures 2 and 3), a method we call "Learning Factors Analysis" goes a step further. Rather than simply visually inspecting learning curves for "blips" like those shown in Figure 2, we can automatically test whether including (or excluding) factors, like first vs. second parameter, leads to better fitting learning curves. Better fits mean a cognitive model that better characterizes what is hard for a student what factors do or do not change how well one practice opportunity transfers to another (e.g., the 5th to 6th opportunity in Figure 2). Croteau, Heffernan, and Koedinger (2004) used Learning Factor Analysis to evaluate alternative models of algebra symbolization.

3. Methodology

3.1 Base Model

The base cognitive model used in our study is the model for the Area unit of Cognitive Tutor Geometry. It has 12 discrete skills -- circle-area, circle-circumference, circle-diameter, compose-by-addition, compose-by-multiplication, equi-tri-height, parallelogram-area, pentagon-area, rectangle-area, square-area, trapezoid-area, and triangle-area. It was generated from the third author's analysis of geometry textbooks and an attempt to simply the original cognitive model he designed for this unit.

3.2 Data acquisition and pre-processing

The data set was extracted from log files for students who used the tutor in their Pittsburgh classroom. The data has four columns – student, success, step, skill. Student is the names of the students. Step is the particular step in a tutor problem the students are involved in. Success is whether the student did that step correctly or not. Skill is the particular skill used in that step. The whole data set has 5431 data points involving 59 students, and 139 problem steps.

3.3 Difficulty Factors

A factor is a hidden feature in a problem, which either makes the problem easier to solve or difficult to solve. It is usually found by theory and task analysis. Suppose the student is asked to find the areas of the following two circles (figure 4), given their radius. The production rule used in this problem is CIRCLE-AREA, i.e. given the radius r, compute the area S = π r2. Although the two problem requires the same production rule, some students may find it easier to solve the first one than the second one. The only difference between them is that the second circle is embedded in a square while the first circle is presented alone.

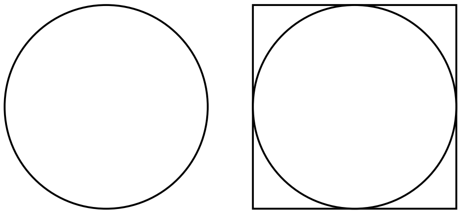

Figure 4 a hypothetic problem

Thus, we can hypothesize that it is the embededness of the circle that causes the difficulty. We code it as a factor called "Embed" with two possible values "embed" or "alone". For notation purpose, the first letter of the factor name is capitalized and all the values are lower cased. When the student encounters a problem with the embedded circle, we record the presence of the factor "Embed" with value "embed". When the student encounters a problem with a single circle, we record the presence of the factor "Embed" with value "alone".

Other factors we found in the geometry lesson are "Repeat", "Backward", "BaseOrHeight", "FigureType", "FigurePart". Their values are listed in table 1.

"Repeat" means whether the production rule to be used is in its first trial. In the given example, if it is the first time for the student to use the production rule CIRCLE-AREA, the factor has value "initial". Otherwise, it has value "repeat".

"Backward" means whether the production rule to be used is in its backward form, rather than the forward form taught. Imagine the student is taught how to compute the area of a circle using production rule CIRCLE-AREA S = π r2. While in the new problem, he is asked to compute the radius, given the area. The production rule to be used is in the backward form. Thus, the factor has value "backward" in this step.

"FigureType" refers to the major geometric type in the problem. It has nine values, eight of which refer to specific geometric type in the step, and one with value zero denoting none previous figure applicable. In the previous circle problem, the FigureType has value "circle".

"FigurePart" refers the part of the figure to be computed. It has twelve values, eleven of which refer to specific part of the figure type in the step, and one with value zero denoting none previous part applicable. The previous problem asks the student to compute the circle area, and thus the FigureType has value "area".

3.4 skill orders and model operators

Skill orders refer to the amount of times a particular skill is used for the same student. It increments every time the skill is used by the same student. Table 2 shows that student A used skill "Rectangle-area" in the first step and in the second step. Thus, the skill order for this skill in the second trial is 2. . Notice that the skill orders are calculated per person. Although the last skill order for "Rectangle-area" for student A is 2, the skill order for the first time use of "rectangle-area" by student B is 1 since B is a different student.

A model operator is a mutation on a base model and generates several submodels by incorporating a factor. We implemented three model operators – partial split, add, and merge.

When a base model A is partial split on a skill by an n-valued factor, that skill is possibly split into n new skills with the element of the factor. For example, table 1 shows that for student A skill "rectangle-area" is used in step 1 and 2 and factor "embed" is has value "alone" in step 1 and "embed" in step 2. Shown in table 2, after partial splitting skill "rectangle-area" on factor "Embed", we get two new skills "rectangle-area-embed" and "rectangle-area-alone". The skill order is recomputed every time an operator is performed on a model. Note worthily, the student has the first time to use rectangle-area-Embed in step 2 in the new model while it was her second time to use rectangle-area in step 2.

Operator "Add" means that the factor with its possible value is simply added as a new skill to the original model. If we add "Embed" to model A, we will have one more skill called "Embed" while the rest skills remain unchanged. Table 4 shows the result after adding "Embed".

Operator "Merge" replaces all the skills with the factor value when the factor value is present. If we merge the base model according to factor "Embed", we will end up with the following skills and skill orders (table 5).

We name the submodel with the names of the all operations it has taken. E.g [add Embed], [merge FigurePart, add BaseOrHeight].

4. Model Search

Given the base model, student performance data, defined factors, and the three operators, we can search a model space to find a model that better accounts for student performance data. We implemented an A* search algorithm, an informed graph search algorithm guided by a heuristic, to search through the gigantic model space. The base model is partial split, added, and merged on all the factors and generates a list of submodels. Each of the submodels is then split, added, and merged. We also added model checking function in the search algorithm to recognize equivalent models to avoid duplicates. Figure 5 shows part of the search space.

To limit the tree size and avoid out of memory problem, after each expansion, we only store the best 10 – 20 submodels according to the heuristic. The trade-off is the optimality vs. memory. By pruning low quality submodels along the search process, we can search a deeper level before the program consumes all memory.

4.1 Multiple Logistic Regression

Multiple regression is a method to study the relationship between a response variable Y and a group of predictor variables. Logistic regression is a type of multiple regression where the dependent

Table 3 Learning Factors

Factor Names	Factor Values				
Embed	alone	embed			
Repeat	initial	repeat			
Backward	forward	backward			
BaseOrHeight	0	Base	Height		
FigureType	0 parallelogram	triangle pentagon	square circle	rectangle segment	trapezoid
FigurePart	0 radius side	area area-combination segment	area-difference base	circumference height	diameter apothem

Table 2 Skills in the Base Model

Student	Step	Skill	Skill Order	Factor - Embed
A	step1	Rectangle-area	1	alone
A	step2	Rectangle-area	2	embed
A	step3	Square-area	1	alone
B	step1	Rectangle-area	1	alone
B	step2	Compose-by-addition	1	embed
B	step3	Compose-by-addition	2	embed

Table 3 Skills in the New Model by Splitting the Base Model on Factor “Embed”

Student	Step	Skill	Skill Order
A	step1	Rectangle-area-alone	1
A	step2	Rectangle-area-embed	1
A	step3	Square-area	1
B	step1	Rectangle-area	1
B	step2	Compose-by-addition	1
B	step3	Compose-by-addition	2

Table 4 Skills in the New Model by Adding Factor “Embed”

Student	Step	Skill	Skill Order
A	step1	Rectangle-area	1
A	step1	Embed - alone	1
A	step2	Rectangle-area	2
A	step2	Embed - embed	1
A	step3	Square-area	1
A	step3	Embed - alone	2
B	step1	Rectangle-area	1
B	step1	Embed - alone	1
B	step2	Compose-by-addition	1
B	step2	Embed - embed	1
B	step3	Compose-by-addition	2
B	step3	Embed - embed	2

Table 5 Skills in the New Model by Mergine Factor “Embed”

Student	Step	Skill	Skill Order
A	step1	Embed - alone	1
A	step2	Embed - embed	1
A	step3	Embed - alone	2
B	step1	Embed - alone	1
B	step2	Embed - embed	1
B	step3	Embed - embed	2

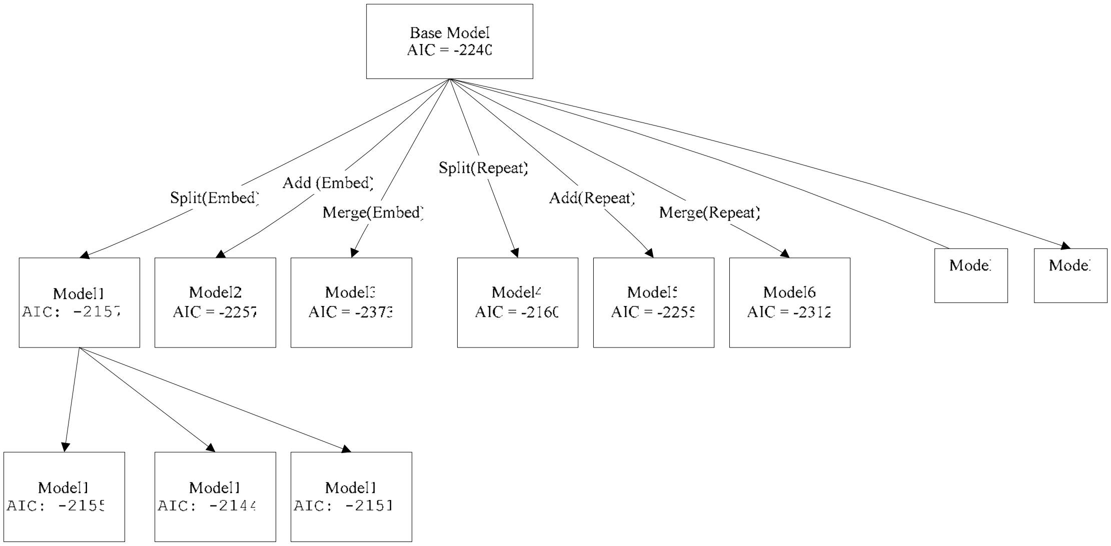

Figure 5 Search Through a Model Space

Heuristic	**Better Model**	**Number of Skills**	**Base model Heuristic value**	**Better model heuristic value**
R square	add FigurePart, add Backward	25	0.149	0.174
Log likelihood	add FigurePart, partial split Embed on circle-area	24	-2,638	-2,572
AIC	add FigurePart, partial split Embed on circle-area	24	5,440	5,353
BIC	merge FigureType, partial split Embed on FigureType.trapezoid	11	5,981	5,943

Table 6 Better models generated by different heuristics

variable has a Bernoulli distribution with the probability of p, and the response variables have either a discrete distribution or continuous distribution. For a k-dimensional predictor variable X, the logit model solves

$$\ln[p/(1-p)] = \beta_0 + \Sigma\beta_j x_{ij}$$

where x_{ij} is the the i-th observation value of a independent variable.

In our method, the dependent variable is whether the student did the step correctly or not. Since we are estimating a different baseline error rate for each individual, we accounted for the dependence by including the student as one of the predictor variables. The other predictor variables are skill, and the interaction between skills and skill orders. The regression formula is

$$\ln[p/(1-p)] = \beta_0 + \alpha_j + {}_{\Sigma\beta j}x_{ij} + \Sigma\gamma_j x_{ij} t_{ij}$$

α_j – the student

x_{ij} - the skill, 1 for being present, 0 for being absent

t_{ij} – the ith skill order for the jth student

$x_{ij}t_{ij}$– the interaction between the skill and the skill order

4.2 Model Selection, R square, Log likelihood, AIC and BIC

What would be a good heuristic in guiding the search? We considered four candidates in our study -- R square, Log likelihood, AIC and BIC. R square, Log likelihood measures the fit between the data and the model. AIC and BIC measure the balance between the fit and the model complexity.

In multiple regression when we have more predictor variables, the bias of the prediction decreases and the variances increases. We need to trade off fit and complexity. We used a commonly used criterion in evaluating our regression model -- AIC) (Akaike Information Criterion)

AIC = -2*log-likelihood + 2*number of parameters

When comparing fitted objects, the smaller the AIC, the better the model is.

Another commonly used evaluator is BIC(Bayesian Information Criterion).

BIC = -2*log-likelihood + number of parameters * number of observations

Both AIC and BIC are asymptotically optimal and consistent. When the true model is within the candidate family of regression functions, BIC will eventually select the true model. However if the true model is not within the candidate family, AIC asymptotically selects the model with the smallest squared error (Yang 2003).

4.3 Implementation

We used Java to implement the model operators and the A*search algorithm, and Splus 6.2 (Splus, 2001) to compute the logistic regression and the heuristics

5. Results

5. 1 Better Models

We experimented with each of these heuristics over the whole dataset. As the huge search space, we limited our search within two tiers. Table 6 presents summarized results. Different heuristic finds different better model. All the heuristic found adding "figure part" improve the base model. Log likelihood and AIC both found adding "figure part" as a separate skill and splitting "embed" on skill "circle-area" a better model than the base model. All the better models found by the heuristics except BIC increased the number of skills.

5.2 Evaluating the models, the operators, and the heuristic

A better cognitive model could be interpreted in several ways. 1) It has better predictive power. It terms of the learning curve, each skill should have a smoother curve. 2) It is parsimonious. According to Occam's razor, a simpler theory is preferred to the more complex. A cognitive model is no exception. We may not always achieve both of them. Thus, the balance of them becomes necessary.

Among the three operators, partial split, add increases the number of skills while merge decreases it. R square, Log likelihood measures the fit while AIC and BIC measure the balance between the fit and the complexity.

Putting more severe penalty for complexity, BIC leads to a smaller model than the other methods (Wasserman, 2004). Not surprisingly, the better model found by BIC is more parsimonious than the base model. The model found by R square and log likelihood is more complex.

6. Conclusion

We are developing a system that can rapidly and automatically evaluate and improve a cognitive model. The first purpose is to reveal the hidden factors in a cognitive model through statistical analysis and model search. The second purpose is to make it easier to refine an existing cognitive model. Currently, the system is able to generate a statistically better model by incorporating the learning factors into the base model. However, the meaning of the better model still needs to be investigated. At this point, it is not clear whether we need to replace the base model with the better model generated by the system. In investigate this issue, we will further refine the system, search in a wider space, and design better heuristics.

Acknowledgement

This research is sponsored by an NSF grant to the Pittsburgh Science of Learning Center, a joint research center located at the University of Pittsburgh and Carnegie Mellon University.

References

Corbett A.T., Anderson, J.R., O'Brien A.T.(1993) Student Modelling in the ACT Programming Tutor, *Cognitively Diagnostic Assessment*, Hillsdale, NJ: Erlbaum

Croteau E.A., Heffernan N. T., Koedinger K.R., (2004) Why are Algebra Word Problem Difficult? Using Tutoring Log Files and the Power Law of Learning to Select the Best Fitting Cognitive Model, Proceedings of Intelligent Tutoring Systems 2004

Freyberger J. (2004) Using Association Rules to Guide a Search for Best Fitting Transfer Models of Student Learning, Master Thesis, Worcester Polytechnic Institute.

Junker B.W., Koedinger, K,, Trottini M.,(2000). Finding Improvements in Student Models for Intelligent Tutoring Systems via Variable Selection for a Linear Logistic Test Model. ,presented at ann mtg of Psychom Soc, 2000, Vancouver BC,

Koedinger K. R (2000), Research Statement for Dr. Kenneth R. Koedinger, June 2000, http://pact.cs.cmu.edu/koedinger/koedingerResearch.html

Koedinger K. R., Mathan S. (2004) Distinguishing Qualitatively Different Kinds of Leaning Using Log Files and Learning Curves, Proceedings of Intelligent Tutoring Systems 2004

Koedinger K.R., Anderson, Hadley & Mark (1995). Intelligent Tutoring Goes to School in the Big City. Proceedings of the 7th World Conference on Art, Intelligence and Education, AACE.

Splus (2001), Splus 6 for Windows Users' Guide, Insightful Corperation.

Yang, Y. (2004b) Can the strengths of AIC and BIC be shared? -A confliict between model identification and regression estimation, manuscript.

Wasserman L.(2004) All of Statistics, 1st edition, Springer-Verlag New York, LLC