Generalized Learning Factors Analysis: Improving Cognitive Models with Machine Learning

by

Hao Cen

A Thesis Submitted to the

Faculty of Carnegie Mellon University

in Partial Fulfillment of the

Requirements for the degree of

DOCTOR OF PHILOSOPHY

Major Subject: Machine Learning

Approved by the
Examining Committee:

_____
Kenneth Koedinger, Thesis Adviser

_____
Brian Junker, Member

_____
Geoff Gordon, Member

_____
Noel Walkington, Member

Carnegie Mellon University
Pittsburgh, Pennsylvania

May, 2009
(For Graduation May 2009)

The copyright page is optional: delete this page if you don't want it.

CONTENTS

Generalized Learning Factors Analysis: i

Improving Cognitive Models with Machine Learning i

CONTENTS iii

LIST OF TABLES vi

LIST OF FIGURES viii

ACKNOWLEDGMENT ix

ABSTRACT x

1. Introduction 1

    1.1 The Challenge of Evaluating and Improving Cognitive Models 1

    1.2 Research Questions and Thesis Overview 2

        1.2.1 Research Questions 3

        1.2.2 Thesis Organization 3

2. Related Work 4

    2.1 Cognitive Psychology 4

    2.2 Psychometrics 5

    2.3 Machine Learning and Data Mining 5

3. Learning Factors Analysis – The Static Part 8

    3.1 The Q-Matrix 8

    3.2 The Additive Factor Model (AFM) 8

    3.3 The conjunctive factor model (CFM) 9

    3.4 Parameter estimation 9

        3.4.1.1 Maximum Likelihood Estimation 9

        3.4.1.2 Penalized Maximum Likelihood Estimation 12

    3.5 Assessment of the Statistical Models 12

    3.6 Assessment of the Cognitive models 13

4. Learning Factors Analysis – The Dynamic Part 14
4.1 The P-Matrix 14
4.2 Model operators 14
4.3 Model search 15
5. Data Analysis with LFA 17
5.1 Simulated Data 17
5.2 Geometry Area – Single Skill per Step 17
5.2.1 Experiment 1 18
5.2.2 Experiment 2 18
5.2.3 Experiment 3 20
5.2.4 Combining the results from experiment 1, 2, 3 21
5.3 EAPS – Conjunctive Skill per Step 22
6. Applications of LFA 25
6.1 Improving Student Learning Efficiency by Reducing Over Practice 25
6.1.1 Discover Learning Inefficiency through LFA 25
6.1.2 Experiment by Tuning the Parameters in the Tutor 27
6.1.3 Saving Student Learning Time while Maintaining Learning Gains 28
6.2 Explain Trading Volume and Skills 29
6.2.1 The Market Data 30
6.2.2 Applying LFA 31
7. Automatic Discovery of Q Matrices 35
7.1 Partial Ordering Knowledge Structure (POKS) 35
7.1.1 Motivation and the Data Set 35
7.1.2 Introduction to POKS 35
7.1.3 Results from POKS 37
7.2 Exponential-Family Principle Component Analysis (EPCA) 39
7.2.1 Principle Component Analysis (PCA) 39

7.2.2 Exponential-Family Principle Component Analysis (EPCA) 39

7.2.3 Application of EPCA for Automatic Discovery of Q Matrices 40

7.2.3.1 Formulation 1 -- directly apply Bernoulli EPCA to student-item matrices (baseline) 40

7.2.3.2 Formulation 2 -- directly apply Bernoulli EPCA to student-item matrices and add student parameter and one 1s row to the skill-step matrix 41

7.2.3.3 Formulation 3 -- directly apply Bernoulli EPCA to student-item matrices, add student parameter and one 1s row to the skill-step matrix, and constrain beta and q 41

7.2.3.4 Formulation 4 -- directly apply Bernoulli EPCA to student-item matrices, Add student parameter and one 1s row to the skill-step matrix, and Constrain beta and q more 42

7.2.4 Complications of applying EPCA to EDM 42

7.2.5 Evaluation of EPCA 42

7.2.6 Results 44

7.2.6.1 Simulated Data 44

7.2.6.2 EPCA vs. LFA on EAPS Data 47

7.2.6.3 What caused over fitting in EPCA? 48

8. Conclusions and Future Work 50

9. Bibliography 51

LIST OF TABLES

Table 1 compares the LFA method with earlier approaches. 6
Table 2 A sample Q-matrix 8
Table 3 Skills and predicted probability for three algebra items 9
Table 4 A Q-matrix 14
Table 5 A P-matrix 14
Table 6 Model statistics after the split 15
Table 7. The sample data 17
Table 8. 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. 18
Table 9. Top three improved models found by LFA with BIC as the heuristic. The table shows the history of splits and model statistics. 18
Table 10. Success probabilities of CMarea and CMsegment 19
Table 11. Success probabilities of CAalone and CAembed 19
Table 12. Top three improved models found by LFA with BIC as the heuristic. 20
Table 13. Statistics of Compose-by- Multiplication before and after split 21
Table 14 Skill coding used in this paper 22
Table 15 A sample of the Q-matrix in the EAPS data 22
Table 16 Empirical success rate for skills 22
Table 17 Model comparison of the EAPS data. The skills are listed in the order of S, H, U. 23
Table 18 Knowledge tracing parameters used in the 1997 Cognitive Geometry Tutor .. 26
Table 19 Time cost in the six tutor curriculum units. The time is in minutes. 29
Table 20 Parameter fits for the securities 32

Table 21 Average training error and cross validation error in each of the three experiments 45
Table 22 Two Q matrices from EPCA on the same data 47
Table 23 Cross Validation Errors by EPCA and LFA on the EAPS data 48

LIST OF FIGURES

Figure 1 A student response model 2
Figure 2 Item response model..... 6
Figure 3 Single latent variable response model 6
Figure 2 A best-first search through the cognitive model space 16
Figure 3 Predicted success rates by skill combinations. 24
Figure 4 Learning Curve of Rectangle-Area and Trapezoid-Area – The solid lines are the actual error rates over the ordered number of practices. The dotted lines are the error rates predicted by LFA..... 26
Figure 5 Pretest and post test scores over the two conditions (left) and the retention test scores (right) 28
Figure 6 Percentage of Time Saved..... 29
Figure 7 Prices of two securities..... 31
Figure 8 Trading volumes of the two securities 31
Figure 9 Learning curves in trial 1 period 3. The solid lines are the actual success rates of using each skill across the ordered number of practices. The dotted lines are the success rates predicted by the LFA model. 33
Figure 10 Density plots for the trader parameters of market makers vs. market takers .. 33
Figure 11 Density plots for the trader parameters of askers vs. buyers, bidders vs. sellers 34

ACKNOWLEDGMENT

Type the text of your acknowledgment here.

ABSTRACT

In this thesis, I propose a machine learning based framework called Learning Factors Analysis (LFA**)** to address the problem of discovering a better cognitive model from student learning data. This problem has both significant real world impact and high academic interest. A cognitive model is a binary matrix representation of how students solve domain problems. It is the key component of *Cognitive Tutors*, an award-winning computer-based math curriculum that grows out of the extensive research in artificial intelligence at Carnegie Mellon. However, discovering a better matrix representation is a structure learning problem of uncovering the hidden layer of a multi-layer probabilistic graphic model with all variables being discrete.

The LFA framework we developed takes an innovative machine learning process that brings human expertise into the discovery loop. It addresses four research questions that one builds upon its predecessor. Accordingly, four techniques are developed to solve each problem.

The first question is how to represent and evaluate a cognitive model. We introduced the concept of Q-matrix and developed a pair of latent variable models – Additive Factor Model and Conjunctive Factor model -- that predict student performance by student prior knowledge, task difficulty and task learning rates.

The second question is how to bring human expertise into the discovery of the latent skill variables. We introduced a technique for human labeling latent factors and developed three graph operators – add, merge and split to incorporate the latent factors in the existing graphical structure.

The third question is how to improve a cogntive model given extensive human labeling. We introduced the concept of P-matrix and developed a penalized A* search built on top of the latent variable models. The search mechanism semi-automatically improves existing cognitive models from human labeling of a new binary matrix. The penalty imposed on the search criteria helps to avoid over fitting the data.

The fourth question is how to automate the latent variable discovery process without human involvement. We developed a binomial version of Exponential Principle Component Analysis that decomposes student-task matrix into a student-skill matrix and a skill-item matrix. We then compared its performance with human labeling.

At the end of the thesis, we discuss several applications of LFA ranging from classroom learning to artificial trading competition. In the classroom setting, we applied LFA to student learning data and used an LFA-improved cognitive model to save students 10% - 30% learning time across several units in a curriculum without hurting their learning performance. The company that markets Cognitive Tutor has started to use improved cognitive models for the 2008 version of the products onward. The estimated timesaving for all U.S. students total is more than two million hours per year. We also applied LFA to study human trading using high frequency trading data from an experimental trading competition. My results showed how investors' prior information could be linked to trading price, volume formation.

1. Introduction

1.1 The Challenge of Evaluating and Improving Cognitive Models

Of all the initiatives to improve the math level of U.S. students, vastly improving K-12 math education has been a top priority. One major development toward this end is Intelligent Tutoring Systems. The technology that drives intelligent tutoring systems is grounded in research into artificial intelligence and cognitive psychology, which seeks to understand the mechanisms that underlie human thought, including language processing, mathematical reasoning, learning, and memory. As students attempt to solve problems using these tutoring systems, the programs analyze their strengths and weaknesses and on that basis provide individualized instruction. Intelligent tutoring systems do not replace teachers. Rather, they allow teachers to devote more one-on-one time to each student, and to work with students of varying abilities simultaneously. They allow teachers to design assignments targeted to individual student needs, thereby increasing student advancement at a better rate.

A primary example of Intelligent Tutoring Systems helping U.S. children learn math is *Cognitive Tutors*, an award-winning computer-based math program that grows out of the extensive research in human learning and artificial intelligence at Carnegie Mellon. Evidence indicates that students using the *Cognitive Tutors* program perform 30% better on questions from the TIMSS assessment, 85% better on assessments of complex mathematical problem solving and thinking, and attain 15-25% higher scores on the SAT and Iowa Algebra Aptitude Test. The equivalent learning results hold for both minority and non-minority students [1-3]. By 2007, more than 500,000 middle school students began using *Cognitive Tutors* across the United States.

The full potential of ITS has not yet been reached, though. The issues mainly concern the efficiency level of the cognitive models used, which is at the heart of most tutoring programs. These models describe a set of math skills that represent how students solve math problems.

With cognitive models, ITS assesses student knowledge systematically and presents curricula tailored to individual skill levels and generates appropriate feedback for students and teachers. An incorrect representation of the domain skills may lead to erroneous curriculum design and negatively affect student motivation. An inaccurate model may waste limited student learning time, and teacher instructional energies, both of which are vital to full achievement. According to Carnegie Learning, teachers reported that many students could not complete the tutor curriculum on time. This issue is serious. First, if students cannot complete the cognitive tutor curriculum, they are likely to fall behind their peers. Second, schools today are calling for increased instruction time to ensure adequate yearly progress. The reality, however, is that students have a limited amount of total available learning time, and teachers have a restricted amount of instructional time. Saving one hour of instructional time can be far more productive than increasing instruction by the same amount. This saved time does not reduce student or teacher workloads, but simply makes better use of the energy and attention given to this subject, thus allowing for greater devotion to other academic areas, thus increasing performance in those subjects. The learning gain may be remarkable.

Getting the appropriate cognitive model is challenging because:

1) There are hundreds of skills involved in a single sub-domain of math. For example, the middle school geometry curriculum is estimated to have over 200 individual skills.

2) Many math skills are not explicitly stated in textbooks, and textbook authors often expect students to acquire those skills via problem solving.

3) Skill is not directly observable. The mastering of a skill can only be inferred from student performance on tasks that require those skills.

4) Initial cognitive models were written by math experts. Many prior studies in cognitive psychology have shown that experts often make false predictions about what causes difficulty for students due to "expert blind spots" [4-11]

The existing cognitive models are usually an incomplete representation of student knowledge, resulting in both less accurate assessment of student knowledge and lower student learning efficiency than desired. Improving the existing cognitive models, given the rate at which the Cognitive Tutor is used across the U.S., has an immediate and significant impact on student learning, and has a long-term impact on transforming math curriculum design. Now, an increasing number of student learning data becomes available. Within Pittsburgh Science of Learning Center, a central education data warehouse has hosted over 50 student learning data sets ranging from the domain of algebra to foreign language learning. The challenge, then, is how do we get a better cognitive model, using student learning data?

1.2 Research Questions and Thesis Overview

A cognitive model is a binary matrix representation of how students solve domain problems. Discovering a better matrix representation is a structure learning problem of uncovering the hidden layer of a multi-layer probabilistic graphic model with the extra complication of all variables being discrete. Figure 1 shows a visual representation of the problem. On the right hand side are the items. Each item has responses as 1 for correct and 0 for incorrect.

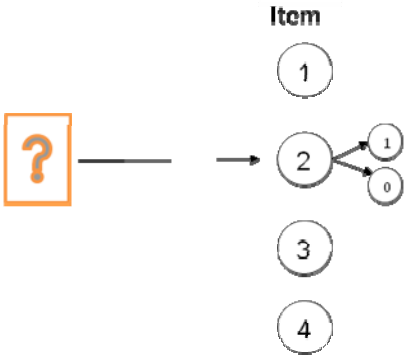

Figure 1 A student response model

Rather than simply pursuing high predictive ability, a unique requirement of this problem is that the interpretability of the cognitive model to human being. After all, these models are used to explain and trace student mastery. Both students and teachers need to be able to understand what the weaknesses and strengths of a student's mastery

of a subject. Being able to communicate its meaning to human is a crucial step for its being to be used by human. The unique side of the Learning Factors Analysis framework brings human expertise into the discovery loop.

1.2.1 Research Questions

The general framework of LFA attempts to answer a series of research questions whose answer that build upon each other.

Question 1 – how to represent and evaluate a cognitive model?

Question 2 – how to bring human expertise into the discovery of the latent skill variables?

Question 3 – how to improve a cognitive model given extensive human labeling?

Question 4 -- how to automate the latent variable discovery process without human involvement?

1.2.2 Thesis Organization

The thesis is organized to answer the research questions

Chapter 2 – We provide an overview of relevant work in machine learning, psychometrics and cognitive psychology.

Chapter 3 – We discuss the concept of Q matrix, a binary matrix representation of a cognitive model. Then we present the set of latent variable models used in LFA – Additive Factor Model and Conjunctive Factor model, their parameter estimation method and evaluation methods. This chapter attempts to answer question 1.

Chapter 4 – We discuss the concept of P matrix, and expert labeling. Heuristic combinatorial search are then presented and three model operators on incorporating the information of P matrix into Q matrix. This chapter attempts to answer question 2 and 3.

Chapter 5 – We applied LFA on real world data sets and how it works.

Chapter 6 -- We applied LFA on real world data sets and show how LFA can be used to answer different research questions. This chapter addresses question.

Chapter 7 – We present several other methods to automatically extract Q matrix and compare their properties with the human guided approach. This chapter attempts to answer question 4.

Chapter 8 – We point out the pros and cons of various approaches in discovering cognitive models and conclude with future work.

2. Related Work

LFA draws strengths from different fields. In machine learning and artificial intelligence, it uses combinatorial search [12], latent factor model [13]. In data mining, it borrows the idea of improving Q-matrix from [14]. In statistics, particularly a branch of statistics called psychometrics, it shares strength with Q-matrix [15] and item response models [16-18]. In cognitive psychology, it extends the early work in learning curve analysis [19]. LFA seamlessly putting the ideas from different field into one framework and in some of those fields, LFA makes a unique contribution and extension. In the following parts, I explain each component in details.

2.1 Cognitive Psychology

The quantitative exploration of finding better cognitive models can be traced back to Newell and Rosenbloom. They found a power relationship between the error rate of performance and the amount of practice [20]. 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".

$$Y = aX^b \tag{1}$$

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

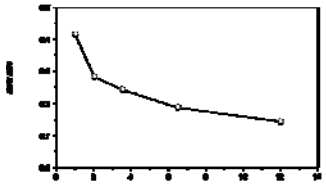

Fig. 1. A power law learning curve

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 [19], 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.

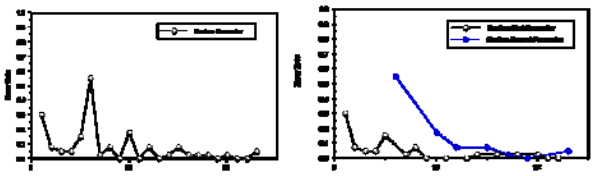

Fig. 2. A learning curve with blips (*left*) split into two smoother learning curves (*right*)

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.

2.2 Psychometrics

Psychometrics is a branch of statistics that is dedicated to psychological assessment, where cognitive models are usually referred to as Q matrices [15].

Item response models [21] apply statistical models to test data to measure test takers' latent traits, such as aptitudes and abilities. Extensions of classic IRT models incorporate information of the skills required by the test items [17, 22-24].

2.3 Machine Learning and Data Mining

In machine learning and data mining, several innovative approaches have been taken to refine an existing cognitive model by having a simulated student to find incorrect rules and to learn new rules via human tutor intervention [25]], using theory refinement to introduce errors to models incorrect student behaviors [26], and using Q-matrix to discover knowledge structure from student response data [14, 15].

Finding a better cognitive model can be naturally situated in the framework of probabilistic graphic models. The student response data constitute the observed layer of nodes, which stand for item responses. The goal becomes finding the latent layer of nodes, which stand for the latent skills. The links between the nodes from the skill layer to the nodes in the item layer form say how skills contribute student performance on the items.

Two extreme solutions to this problem are 1) to model the student responses with all the items (Figure 2), and 2) to model the student responses with a single latent factor, such as student intelligence, aptitude (Figure 3). These two approaches represent the way that modern educational tests are built upon.

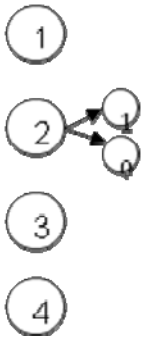

Figure 2 Item response model

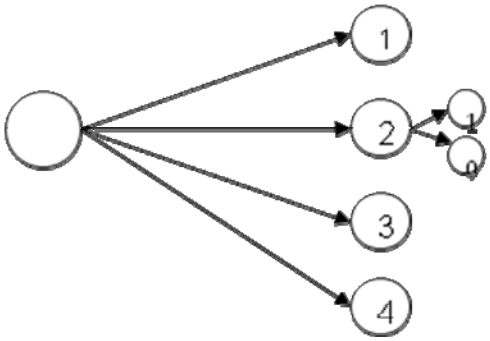

Figure 3 Single latent variable response model

One distinction between LFA and the previous two extremes is that LFA characterizes student responses on items in terms of the skills students uses, i.e. the cognitive model, seen in Figure 4.

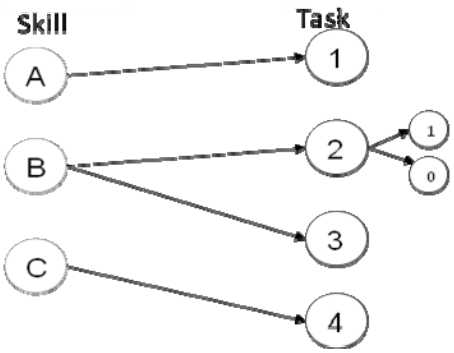

Figure 4 Cognitive model

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.

Table 1 compares the LFA method with earlier approaches.

Task |Features	IRT-based, account for student differences	Handle conjunctive skills	Automatic search for better models	Applicable to learning data	Discover factors that are directly interpretable
DiBello et al.'s	Yes	Yes			

models					
Q-Matrix	Yes	Yes	Yes		
Draney, et al.'s model	Yes			Yes	
Gordon's EPCA		Additive		Yes	Depends
LFA	Yes	Yes	Yes	Yes	Yes

3. Learning Factors Analysis – The Static Part

3.1 The Q-Matrix

Q-matrix is a boolean matrix describing the relationship between items and skills [15] . A cell value of 1 at the row i, column j means that the item i requires the use of skill j. A cell value of 0 means otherwise. Table 2 shows such a relationship between two testing items and four associated skills. Notice the first item requires only one skill and the second item requires two skills simultaneously. Requiring multiples skills simultaneously is called conjunctivity of skills.

Table 2 A sample Q-matrix

Item | Skill	Add	Sub	Mul	Div
2*8	0	0	1	0
2*8 - 3	0	1	1	0

3.2 The Additive Factor Model (AFM)

Modeling the item response given the skills requires a statistical model. The first model we developed, depicted by equation (2), capture that the probability for student i to get item j right is proportional to required knowledge the student knows plus "easiness" of that skill and the learning acquired through practice,

$$p_{ij} = \Pr(U_{ij} = 1 \mid \theta_i, \beta_k, \gamma_k) = \frac{\exp(\theta_i + \sum_{k=1}^{K} q_{jk}\beta_k + \gamma_k T_{ik})}{1 + \exp(\theta_i + \sum_{k=1}^{K} q_{jk}\beta_k + \gamma_k T_{ik})} \qquad (2)$$

where

U_{ij} = the response of student i on item j

θ_i = coefficient for student i

β_k = coefficient for skill k

γ_k = coefficient for the learning rate of skill k

T_{ik} = the number of practice opportunities student i has had on the skill k

$$q_{jk} = \begin{cases} 1 & \text{item j uses skill k} \\ 0 & \text{otherwise} \end{cases}$$

The term "Additive" comes from the linear combination of skill ks in item j in the exponent. That is, if an item requires multiple skills, this model will use the linear combination of the item parameters to predict the overall response.

The model has a connection with Logistic Regression by modeling success as a Bernoulli distribution with the probability of p, and student intelligence, skill easiness, and learning as predictors.

This model also has a connection with Item Response Theory. The additive factor model without the learning term reduces to the Linear Logistic Test Model [23] with skills as the item attributes.

3.3 The conjunctive factor model (CFM)

One problem with AFM is the way it handles conjunctive skills. Suppose there is an item requiring two skills, as shown in Table 3. Assume a student has a theta value of 0; two skills above have beta values logit(.8) and logit(.5) (which is 0); and there are no learning ($\gamma = 0$). In a conjunctive sense, we need a prediction of .4 (= 1/(1 + exp(-(logit(.8))) * 1/(1 + exp(-(logit(.5))) = .8 * .5). AFM will predict the third item with probability of .8 (=1/(1 + exp(-(logit(.8) + logit(.5)))), making a harder item easier.

Table 3 Skills and predicted probability for three algebra items

Item	Skill	P
2*8	mult	.8
7 - 3	sub	.5
2*8 - 3	mult, sub	.5 * .8 = .4

The conjunctive factor model (CFM), depicted by equation (3), captures the idea that when an item requires multiple skills present, the item is harder than the items requiring only one of those skills. The parameters in CFM have same meaning as those in AFM. CFM and AFM reduces to the same form when there is only one skill per item.

$$p_{ij} = \prod_{k=1}^{K} \left(\frac{e^{\theta_i + \beta_k + \gamma_k T_{ik}}}{1 + e^{\theta_i + \beta_k + \gamma_k T_{jk}}}\right)^{q_{jk}} \qquad (3)$$

The conjunctive IRT model in equation 2 builds upon Embretson's multicomponent latent trait model (MLTM) [17], Dibello's Unified Model (UM)[16], and Davier's General Diagnostic Model (GDM) [18].

3.4 Parameter estimation

3.4.1.1 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) has good asymptotic properties for estimators. Thus, we used a method called Joint Maximum Likelihood Estimation (JML) to estimate the student, skill, learning parameters all together.

$$[\theta, \beta, \gamma] = \arg\max_{\theta,\beta,\gamma} LogLikelyhood[\theta, \beta, \gamma; \mathcal{X}] \qquad (4)$$

where θ, β, γ are the student, skill, and learning parameter and $\mathcal{X}$ is the data matrix.

The Additive Factor Model can be thought as the Bernoulli case of a linear generalized model, i.e. logistic regression, with

$$Y_i \sim Ber(p_i)$$

$$\log(\frac{p_i}{1-p_i}) = \beta^T \mathbf{x}_i = z_i,$$

$$p_i = \frac{1}{1+e^{-z_i}}$$

where

i, the index of data points, $i = 1 \ldots n$

Y_i, an observation from a Bernoulli random variable with probability p_i

$$Y_i \sim Ber(p_i) \ , \Pr(Y_i = y_i) = P(y_i) = p_i^{y_i}(1-p_i)^{1-y_i}$$

$\boldsymbol{x}_i$, the value of the dependent variables for observation i

$\boldsymbol{\beta}$, the parameters to estimate

Thus the log likelihood of the parameters given the data is

$$Likelihood \quad L(data) = \prod_{i=0}^{n} P(y_i)$$

$$= \prod_{i=1}^{n} p_i^{y_i} (1-p_i)^{1-y_i}$$

$$LogLikelihood \quad ll(data) = \log(L(data))$$

$$= \log(\prod_{i=1}^{n} p_i^{y_i} (1-p_i)^{1-y_i})$$

$$= \sum_{i=1}^{n} (y_i \log p_i + (1-y_i)\log(1-p_i))$$

$$= \sum_{i=1}^{n} (y_i \log(\frac{1}{1+e^{-z_i}}) + (1-y_i)\log(1-\frac{1}{1+e^{-z_i}}))$$

$$= \sum_{i=1}^{n} (y_i \log(\frac{1}{1+e^{-z_i}}) + (1-y_i)\log(\frac{1+e^{-z_i}-1}{1+e^{-z_i}}))$$

$$= \sum_{i=1}^{n} (y_i \log(\frac{1}{1+e^{-z_i}}) + (1-y_i)\log(\frac{e^{-z_i}}{1+e^{-z_i}}))$$

$$= \sum_{i=1}^{n} (y_i \log(\frac{e^{z_i}}{(1+e^{-z_i})e^{-z_i}}) + (1-y_i)\log(\frac{e^{-z_i}e^{-z_i}}{(1+e^{-z_i})e^{-z_i}}))$$

$$= \sum_{i=1}^{n} (y_i \log(\frac{e^{z_i}}{1+e^{z_i}}) + (1-y_i)\log(\frac{1}{1+e^{z_i}}))$$

$$= \sum_{i=1}^{n} (y_i \log(e^{z_i}) - y_i \log(1+e^{z_i}) + (1-y_i)\log(1) - (1-y_i)\log(1+e^{z_i}))$$

$$= \sum_{i=1}^{n} (y_i z_i - \log(1+e^{z_i}))$$

The Conjunctive Factor Model can be viewed as the Bernoulli case of a nonlinear generalized model with

$$z_{rk} = \theta_{ri} + \beta_{rk} + \gamma_{rk} T_{rk} \tag{5}$$

$$p_r = \prod_{k=1}^{K} (\frac{1}{1+e^{-z_{rk}}}) \tag{6}$$

where K is the total number of skills required in the step in data point r

By multiplying p_r we drive the log likelihood in equation(7).

$$l_{MLE} = \log Likelihood = \sum_{r=1}^{n} (y_r \log(p_r) + (1-y_r)\log(1-p_r)) \tag{7}$$

By taking the derivative of the log likelihood function with respect to the parameters, we drive the gradient for each parameter in equation (8), (9) and (10).

$$\frac{dl}{d\theta_i} = \sum_{r=1}^{n} \frac{dl}{dp_r}\frac{dp_r}{d\theta_i} = \sum_{i=1}^{n}\left(\frac{p_r - y_r}{(p_r-1)p_r} \cdot \sum_{k=1}^{K}\left(\frac{1}{1+e^{-z_{rk}}}\right)p_r\right)$$
$$= \sum_{i=1}^{n}\left(\frac{p_r - y_r}{(p_r-1)} \cdot \sum_{k=1}^{K}\left(\frac{1}{1+e^{-z_{rk}}}\right)\right) \quad (8)$$

$$\frac{dl}{d\beta_k} = \sum_{r=1}^{n} \frac{dl}{dp_r}\frac{dp_r}{d\beta_k} = \sum_{i=1}^{n}\left(\frac{p_r - y_r}{(p_r-1)p_r} \cdot \frac{e^{K\theta_i + \sum_{k=1}^{K}\beta_k + + \sum_{k=1}^{K}\gamma_k T_{kr}}}{\prod_{k=1}^{K}(1+e^{z_{rk}})} \cdot \frac{1}{1+e^{z_{rk}}}\right) \quad (9)$$

$$\frac{dl}{d\gamma_k} = \sum_{r=1}^{n} \frac{dl}{dp_r}\frac{dp_r}{d\gamma_k} = \sum_{i=1}^{n}\left(\frac{p_r - y_r}{(p_r-1)p_r} \frac{T_{kr}e^{K\theta_i + \sum_{k=1}^{K}\beta_k + + \sum_{k=1}^{K}\gamma_k T_{kr}}}{\prod_{k=1}^{K}(1+e^{z_{rk}})} \cdot \frac{1}{1+e^{z_{rk}}}\right) \quad (10)$$

By doing an unstrained optimization on the log likelihood function, we can get the student, skill, learning parameters all together from both AFM and CFM.

3.4.1.2 Penalized Maximum Likelihood Estimation

In preliminary work, we found that freely maximizing the likelihood based on Equation often yielded student parameters that appear unreasonable. We hypothesize that it is caused by over fitting. To overcome this drawback, we designed a Penalized Maximum Likelihood Estimation method (PMLE) [27], which penalizes the oversized student parameters in joint maximum likelihood estimation. Thus, PMLE maximizes the penalized likelihood depicted in Equation (11). Maximizing the penalized likelihood in Equation (11) is equivalent to finding a posterior mode for a Bayesian model, with a normal prior on the θ and flat priors on β and γ A higher value for λ below corresponds to lower prior variance.

$$ll_{PMLE} = ll_{MLE} - \frac{1}{2}\lambda\sum_{i=1}^{I}\theta_i^2,\ \lambda\text{=1 by default} \quad (11)$$

where

I, the total number of students

3.5 Assessment of the Statistical Models

Good statistical models balance between model fit & complexity minimizing prediction risk. They captures sufficient variation in data but is not overly complicated [28].

We choose two measures for model assessment -- Stratified K-Fold Cross Validation, shown in equation (12), which is time-consuming and more accurate estimate of prediction errors, and BIC, shown in equation (13) , which can be fast to

compute but may be a crude approximate of the prediction errors. The stratification is taken on the student side.

$$CV = \sum_{i=1}^{n} (Y_i - \hat{f}^{-\kappa(i)}(x_i))^2 \qquad (12)$$

$$BIC = -2LogLikelihood + numParemeter * numObservation \qquad (13)$$

test

3.6 Assessment of the Cognitive models

With a chosen statistical model, we can then proceed to compare various cognitive models. In the EAPS data, we compared the EAP with known conjunctive skills. The results show that the three-skill cognitive model is better than the one-skill model and the item skill model in terms of BIC. This justifies the use of skills to predict item responses.

4. Learning Factors Analysis – The Dynamic Part

4.1 The P-Matrix

Section 4.1, 4.2, 4.3 are the innovative parts of Generalized Learning Factors Analysis to create a better cognitive model. First, corresponding to the Q-Matrix, we propose a new concept call P-Matrix (the Problem Matrix). A Q-matrix is pre-labeled by domain experts before it is put to use by students. A P-matrix is post labeled by domain experts. After domain experts reviewed the student responses data, they may find some items labeled with the same set of skills have various degrees of difficulties. As seen in Table 4, the second and the third item are labeled with the same set of skills. However, the third item is associated with a higher error rate. A further investigation of the item shows that the third item deals with negative numbers, imposing more difficulty for students. Thus, we can create a P-matrix with item as the row and hypothetical difficulty factors as the columns. In this example, we can put "Dealing with negative number" as one difficulty factor. The first two items have zero as the factor value and the third item has 1 as the factor value.

Table 4 A Q-matrix

Item | Skill	Add	Sub	Mul	Div
2*8	0	0	1	0
2*8 – 3	0	1	1	0
2*8 - 30	0	1	1	0

Table 5 A P-matrix

Item | Skill	Dealing with negative number	…
2*8	0	
2*8 - 3	0	
2*8 - 30	1	

4.2 Model operators

The second step to create a better cognitive model is to turn an existing cognitive model, a Q-matrix, and a P-matrix into new models. Model operators perform that function. One model operator is "Split". In the example above, we have a cognitive model with 15 skills. The last of the ten skills is "triangle-side". "Split" splits "triangle-side" into "triangle-side-base" and "triangle-side-height". Now the new model has 16 skills. Since this data set has only single-skilled items, we can run either AFM or CFM, both without the learning term with PMLE on the model and compare their BICs. Shown in Table 6, the new model is not better than the original cognitive model in terms of BIC, suggesting the factor is not necessary.

Table 6 Model statistics after the split

	LL	BIC
Original	-2,003	4,330
After split	-2,000	4,333

The "Split" operator does not change the conjunctivity of the Q-matrix. The other operator "Add" changes the conjunctivity of the Q-matrix. In triangle example above, suppose we have two items – one calculating the area of the triangle given the sides, and the other calculating one side of the triangle, given the area and the other side. Instead of having these two separate skills for different scenarios, researchers can formulate one skill for the triangle area and the other skill as the algebra manipulation, which is to turn the triangle area formula into triangle side formula. To solve the first item, a student only needs the first skill. To solve second item, a student needs to use the first skill and the second skill.

4.3 Model search

If we have several difficulty factors in a P-matrix, we can do the third step in improving a cognitive model. A distinguishing feature of the LFA method is its semi-automatic model search process. We formulated finding a better cognitive model as a combinatorial search problem. Given an existing cognitive model, a Q-matrix, a P-matrix, the LFA method automatically incorporates those factors into models, and finds new models that researchers may wish to investigate further.

The search algorithm in LFA is a best first search [12]. It starts from an initial node, iteratively creates new adjoining nodes, and explores them to reach a goal node. Difficulty factors are incorporated into an existing IRT model through splits or adds. To limit the search space, it employs a heuristic to rank each node and visits the nodes in the order of this heuristic estimate. Measuring both the model fit and the model complexity, Bayesian Information Criterion (BIC) [28] are the heuristics used in the search. As shown in Figure 5, at the beginning of a search with BIC as the heuristic, the original model is evaluated and BIC is computed. Then the model is split into a few new models by incorporating the factors. BICs are computed from each of the new models. The search algorithm chooses the best one (the shaded node with value 4301) for the next model generation. The search algorithm does not always move to a lower level in the search hierarchy. It may go up to select a model (the shaded node with value 4212) to expand if all the new models have worse heuristic scores than the previous model had. After several expansions, it finds a best model with the lowest BIC value within all the models searched.

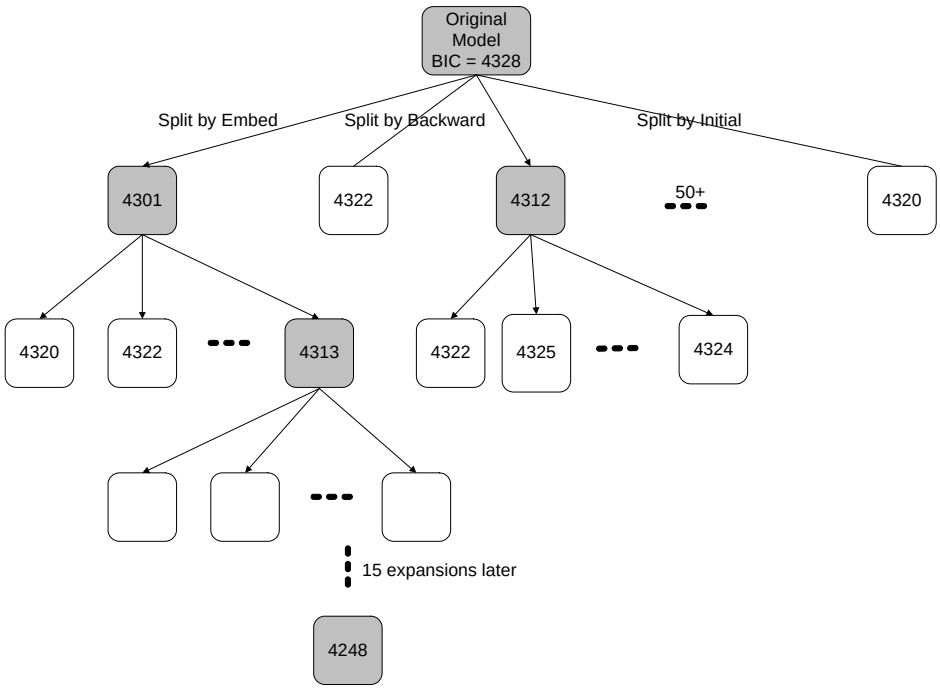

Figure 5 A best-first search through the cognitive model space

5. Data Analysis with LFA

5.1 Simulated Data

5.2 Geometry Area – Single Skill per Step

The 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 7. 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

5.2.1 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 8. 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 Opportuntie s	Initial Probabilit y	Avg Probabilit y	Final Probabilit y
Parallelogra m-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.

5.2.2 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 9. 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 2. Binary split circle-radius by repeat repeat 3. Binary split compose-by-addition by backward backward	1. Binary split compose-by-multiplication by figurepart segment 2. Binary split circle-radius by repeat repeat 3. Binary split compose-by-addition by figurepart area-difference	1. Binary split compose-by-multiplication by figurepart segment 2. Binary split circle-radius by repeat repeat
Number of Skills: 18	Number of Skills: 18	Number of Skills: 17
AIC: 3,888.67 BIC: 4,248.86 MAD: 0.071	AIC: 3,888.67 BIC: 4,248.86 MAD: 0.071	AIC: 3,897.20 BIC: 4,251.07 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 10. 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 11. Success probabilities of CAalone and CAembed

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

5.2.3 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 12. 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 Circle *radius*initial Circle *radius*repeat Compose-by-addition Compose-by-addition*area-difference Compose-by-multiplication*area-combination Compose-by-multiplication*segment	All skills are the same as those in model 1 except that 1. Circle is split into Circle *backward*initial, Circle *backward*repeat, Circle*forward, 2. Compose-by-addition is not split	All skills are the same as those in model 1 except that 1. Circle is split into Circle *backward*initial, Circle *backward*repeat, Circle *forward, 2. Compose-by-addition is split into Compose-by-addition and Compose-by-addition*segment
AIC: 3,884.95 BIC: 4,169.315 MAD: 0.075	AIC: 3,893.477 BIC: 4,171.523 MAD: 0.079	AIC: 3,887.42 BIC: 4,171.786 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 = π*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.

5.2.4 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 13. 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-multiplication. 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.

5.3 EAPS – Conjunctive Skill per Step

The EAPS data is taken from a difficulty factor study of 247 U.S. algebra students in an urban school [10, 29]. Each student had a different quiz with 8 items from an item pool of 96 items. The total number of observations is 1976. The authors of the study characterized the items with a cognitive model involving conjunctive production rules or skills. A simplification of their skill coding involves the following 3 skills, as shown in Table 14. A sample of the Q-matrix is shown in Table 15. Notice certain items, such as "waiter-story-result-easy-mult" has no skills labeled in this Q-matrix due to the simplification.

Table 14 Skill coding used in this paper

Skill Abbreviation	Skill Meaning
S	Symbolic Comprehension -- necessary for reading an equation
H	Arithmetic Procedure Hard (e.g. with decimal numbers like 2.45/7)
U	Unwind Constraint -- necessary for start-unknown or algebra problems, like 7x = 35, but not for result-unknown or arithmetic problems, like 7*5 = x.

Table 15 A sample of the Q-matrix in the EAPS data

	S	U	H
bball-equation-result-easy-div	1	0	0
donut-equation-result-hard-div	1	0	1
lottery-word-start-hard-mult	0	1	1
waiter-story-result-easy-mult	0	0	0
waiter-word-start-easy-div	0	1	0
… …			

The conjunctive skill properties are implied in the empirical success rate over items with the skills in Table 16, as most of the items skill multiple skills are harder that the items with the corresponding individual skills. The last row in this table shows the success rates for the items that were not labeled with the skills in the Q-matrix. It appeared that these items with unlabeled skills are easy items.

Table 16 Empirical success rate for skills

Skill Combination	Success Rate
S	0.75
H	0.63
U	0.61
SH	0.67
SU	0.59
HU	0.43
SHU	0.38
Unlabelled	0.80

Table 17 Model comparison of the EAPS data. The skills are listed in the order of S, H, U.

	CVMean	CVSd	LL	BIC	$\hat{\beta}$ in probability	$\hat{\beta}$ in logit
AFM	0.19	0.27	-832	3560	(0.16, 0.23, 0.19)	(-1.65, -1.23, -1.45)
CFM	0.2	0.29	-1500	4910	(0.9, 0.95, 0.94)	(2.2, 2.87, 2.71)
AFM-P	0.2	0.14	-1130	4150	(0.35, 0.47, 0.43)	(-0.63, -0.14, -0.3)
CFM-P	0.19	0.22	-1660	5230	(0.61, 0.7, 0.67)	(0.43, 0.85, 0.7)

As shown in the model comparison statistics in Table 17, the four combinations have close performance in terms of the cross validation scores. CFM-P has slightly better CVMean. In terms of the skill parameter estimates, CFM-P produces the estimates closest to the empirical single skill success rates shown in Table 16. Notice that $\hat{\beta}$ in logit by AFM and AFM-P are all negative, forcing AFM to behave conjunctively.

In terms of interpretability, CFM-P produces more interpretable skill estimates. Suppose we are to predict the success rate on a new item that requires skill S and H. We can multiple the skill estimates in probability for S and H and get .43 (=.61*.7).

Figure 6 shows the predicted success rates of CFM-P and AFM-P aggregated over the skill combinations. 1) The CFM-P model is almost always closer to the data than AFM-P and it is only more than 10% away for the unlabelled items. AFM-P is more than 10% away from the data in 4 of the 8 cases and is generally more conservative (the predictions are closer to .5 in general and line is flatter). 2) The one particularly bad point for CFM-P is for the 000 row in the Q matrix and suggests a potential modification in the penalty on student parameters. This bad point may cause CFM-P to have a much worse BIC in this comparison.

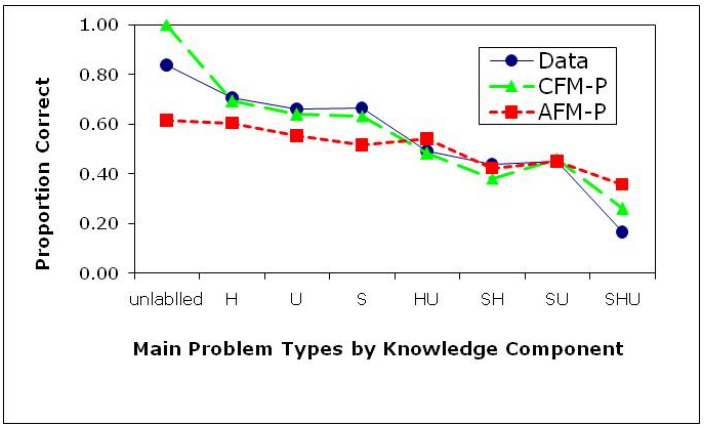

Figure 6 Predicted success rates by skill combinations.

6. Applications of LFA

Many researchers have used LFA as a new research tool to answer questions in domains beyond math and Cognitive Tutors. Researchers Rafferty (Stanford University) & Yudelson (University of Pittsburgh) applied LFA to incorporate learner characteristics and demonstrate that the different student groups require different cognitive models. Their results suggest that by incorporating learner's traits to cognitive models, computer tutors can adapt to students to a greater flexibility and help certain student group achieve higher learning efficiency. Nwaigwe (Carnegie Mellon University) and colleagues at the University of Pittsburgh used LFA to explore the quality of different methods for analyzing student errors during training. Leszczenski (Carnegie Mellon University) and Beck (Worchester Polytechnic Institute) extended LFA to answer a perennial research question on reading transfer: If a child learns to read a word (e.g., "cat"), will that child be able to better learn other related words (e.g., "cats" or "dog")? They used LFA to analyze data from a computerized tutor that listens to children while reading (created through $6 million grant support from the National Science Foundation). They discovered that when children learn to read, there is transfer of learning from word roots to related words – an important finding relevant to national debates about whether early reading instruction should emphasize phonetics or word meanings. LFA was also used to discover that students at higher levels of reading proficiency show greater range of transfer, a result that supports the hypothesis that helping students make connections between words in the same "family" may accelerate the sometimes slow process of learning to read. Foreign language researcher Yanhui Zhang used LFA to investigate whether and to what extent Chinese segmental distinctions and suprasegmental distinctions are perceived differently as a function of linguistic experience.

One application we did with LFA was to use LFA to discovery over practice and underpractice in the tutoring environment with the goal to improve student learning efficiency.

6.1 Improving Student Learning Efficiency by Reducing Over Practice

6.1.1 Discover Learning Inefficiency through LFA

By applying LFA to the student log data from the Area unit of the 1997 Geometry Cognitive Tutor, we found two interesting phenomena. On the one hand, some easy (i.e. high β_j) KCs with low learning rates (i.e. low γ_j) are practiced many times. Few improvements can be made in the later stages of those practices. KC rectangle-area is an example. This KC characterizes the skill of finding the area of a rectangle, given the base and height. As shown in Figure 7, students have an initial error rate around 12%. After 18 times of practice, the error rate reduces to only 8%. The average number of practices per student is 10. Many practices spent on an easy skill are not a good use of student time. Reducing the amount of practice for this skill may save student time without compromising their performance. Other over-practiced KCs include square-area, and parallelogram-area. On the other hand, some difficult (i.e. low β_j) KCs with high learning rates (i.e. high γ_j) do not receive enough practice. Trapezoid-area is such an example in the unit. But students received up to a maximum of 6 practices. Its initial error rate is 76%. By the end of the 6th practice the error rate remains as high as 40%, far from the level of mastery. More practice on this KC is needed for students to reach mastery. Other under-practiced KCs include pentagon-area and triangle-area.

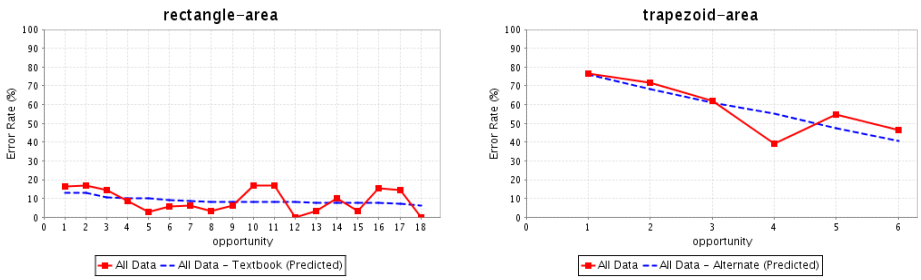

Figure 7 Learning Curve of Rectangle-Area and Trapezoid-Area – The solid lines are the actual error rates over the ordered number of practices. The dotted lines are the error rates predicted by LFA.

What caused the over practice in the Cognitive Tutor curriculum? Cognitive Tutor uses the Knowledge Tracing algorithm to update its estimates of students' mastery of KCs [19]. Based on these estimates, the Tutor chooses to give students the problems with the skills students need to practice more. Table 18 explains the meaning of the four parameters $P(L_0)$, P (T), P(Guess), P(Slip) used in the update. We discovered that the 1997 Tutor used the same set of parameter estimates for all the KCs, as shown in Table 18 column 3. Then we fit the four parameters for each KC with the student log data and used the fit parameters to estimate the amount of practice for these KCs in the same dataset. We found that 58% out of 4102 practices and 31% of 636 exercise questions were done after the students had reached mastery. On the other hand, 119 more practices were needed for all students to master those under-practiced KCs. Although applying the fit parameter estimates on the training data may incur over fitting, the finding does suggest that using a set of carefully calibrated parameter estimates may improve learning efficiency.

To test the effect of calibrated Knowledge Tracing parameters, we planned a study in 2006, when the Geometry Cognitive Tutor had evolved into its 2006 version. The 2006 Tutor breaks the single 1997 area unit into 6 area units (Squares & Rectangles, Parallelograms, Triangles, Trapezoids, Polygons, and Circles), and has a different cognitive model, curriculum design, interface, and student population from its predecessor.

Table 18 Knowledge tracing parameters used in the 1997 Cognitive Geometry Tutor

Parameter	Meaning (The probability that …)	Estimate
$P(L_0)$	the KC is initially known	0.25
P(T)	the KC transit form an unknown state to a known state	0.2
P(Guess)	a student will apply a KC correctly even if the KC is not learned	0.2
p(Slip)	a student will apply a KC incorrectly even if the KC is learned	0.1

53 KCs are specified in the six units. 32 of them involve calculating perimeters and areas of various shapes. 21 of them involve extracting specific numbers from the text questions and entering them into the tutor interface. The numbers of KCs in each of the 6

units are 19, 7, 7, 8, 4, and 6 respectively. The same set of knowledge tracing parameter estimates shown in Table 18 were used for all its KCs.

Because the 2006 Tutor was just released several weeks before our study, there were no student-log data available from that Tutor to evaluate the KCs empirically. The most recent data available were from the 2005 version of the Cognitive Geometry Tutor, which used the same set of knowledge tracing parameter estimates shown in Table 18. Not surprisingly, we found over-practice and under-practice in that tutor. The over-practiced KCs and the under-practiced KCs are slightly different across the tutors.

Because the cognitive model in the 2005 Tutor is slightly different from the model in the 2006 Tutor, we could not exactly copy those parameter estimates into the 2006 Tutor. To approximate the 2006 estimates, we grouped the KCs in the 2006 Tutor into several homogeneous groups according to their degrees of over-practice in 2005. This is a qualitative mapping of the parameters of one tutor version with a different student population to the parameters of another tutor version. Within each group, KCs share the same parameter estimates. Because we had no relevant information on slips or guesses, we mainly focused on adjusting $P(L_0)$ and P(T) in our study. Between groups, KCs vary mainly on $P(L_0)$. If a KC shows a much higher learning rate P(T) than the original parameter estimate .2, we increased P(T) for that KC. Meanwhile, in order to reduce the danger of under-practice, we reduced all the $P(L_0)$ by a certain amount from the fit estimates. The final parameter estimates in the optimized tutor have the following changes.

The under-practiced KC (circle-area) is set to a lower $P(L_0) = 0.2$.

The under-practiced KC with a high learning rate (triangle-area) is set to a lower $P(L_0) = 0.2$, and a higher P(T) = .5.

The slightly-over-practiced KCs (circle-circumference, trapezoid-area, trapezoid-perimeter, triangle-perimeter) are set to $P(L_0) = 0.5$.

The moderately-over-practiced KCs (parallelogram-area, parallelogram-perimeter, rectangle-area, rectangle-length-or-width, rectangle-perimeter, square-area, square-perimeter, square-side-length) are set to $P(L_0) = 0.7$.

All the KCs for information extraction are set to $P(L_0) = 0.9$.

Based on these changes, their locations in the units, and the number of KCs in each unit shown, we made the following research hypothesis. Compared with the students in the control condition, students using the optimized tutor will learn the same amount but spend

much less time in unit 1 and 2 (Squares and Parallelograms);

moderately less time in unit 3 (Triangles);

the same amount of time in unit 4 and 5 (Trapezoids and Polygons)

more time in unit 6 (Circles)

6.1.2 Experiment by Tuning the Parameters in the Tutor

The experiment was conducted in the regular instruction of the geometry course. 110 students from a total of 6 classes in a high school near Pittsburgh participated in the study. They were all taught by the same teacher. The students were randomly assigned to the optimized condition, where they would be using the optimized tutor, or to a control condition where they would be using the original 2006 Tutor with no modifications. We designed a pretest, a post test and a retention test to assess students' ability to solve

geometry area and perimeter problems. The test forms were counter-balanced. Each test has 13 – 14 test items including both regular and transfer items.

In both conditions, the students took the pretest shortly before they started working on the first unit. Then students spent about 3 days per week on regular classroom instruction and 2 days per week using the Tutor in a computer lab. In the lab, students worked on either the optimized tutor or the original Tutor, according to the condition they had been assigned to. Shortly after finishing the 6th unit, they took the post test. In two weeks after each student finished the post test, we gave each student a retention test. Students' interaction and learning time with the tutor were logged. Due to student attrition and recording errors, 94 students took the pretest; 84 students took the post test; 64 students took the retention test; 73 students took both the pretest and the post test and had valid test scores; and 62 students had valid log data.

6.1.3 Saving Student Learning Time while Maintaining Learning Gains

We found that the optimized group learned as much as the control group but in less time. As seen in Figure 8 (left), the two groups have similar scores in both the pre test and the post test. The amount of learning gain in both groups is approximately 5 points. To further examine the treatment effect, we ran an ANCOVA on the post test scores, with condition as a between subject factor, the pretest scores as a covariate, and an interaction term between the pretest scores and the condition. The post test scores are significantly higher than the pretest, $p < .01$, suggesting that the curriculum overall is effective. Meanwhile neither the condition nor the interaction are significant, $p = 0.772$, and $p = 0.56$ respectively. As shown in the Figure 8 (right), we found no significant difference in the retention test scores ($p = 0.602$, two tailed). The results from the post test and the retention tests suggest that there is no significant difference between the two groups on either of the two tests. Thus, over practice does not lead to a significantly higher learning gain.

Figure 8 Pretest and post test scores over the two conditions (left) and the retention test scores (right)

The actual learning time in each unit matches our hypotheses. As shown in Table 19, the students in the optimized condition spent less time than the students in the control condition in all the units except in the circle unit. The optimized group saved the most amount of time, 14 minutes, in unit 1 with marginal significance $p = .19$; 5 minutes in unit 2, $p = .01$, and 1.92, 0.49, 0.28 minutes in unit 3, 4, and 5 respectively. In unit 6, where we lowered $P(L_0)$, the optimized group spent 0.3 more minutes. Notice the percentage of the time saved in each unit. The students saved 30% of tutoring time in

unit 2 Parallelogram, and 14% in unit 1 Square. In total students in the optimized condition saved around 22 minutes, an 12% reduction in the total tutoring time.

Table 19 Time cost in the six tutor curriculum units. The time is in minutes.

	Optimized	Control	Time saved	% time saved	t Stat	P(T<=t) one-tail
Square	87.16	101.18	14.02	14%	-0.89	0.19
Parallelogram	11.83	16.95	5.12	30%	-2.58	0.01
Triangle	13.03	14.95	1.92	13%	-0.91	0.18
Trapezoid	26.39	26.88	0.49	2%	-0.15	0.44
Polygon	10.58	10.86	0.28	3%	-0.18	0.43
Circle	13.42	13.12	-0.30	-2%	0.18	0.43
Total	162.41	183.93	21.52	12%		

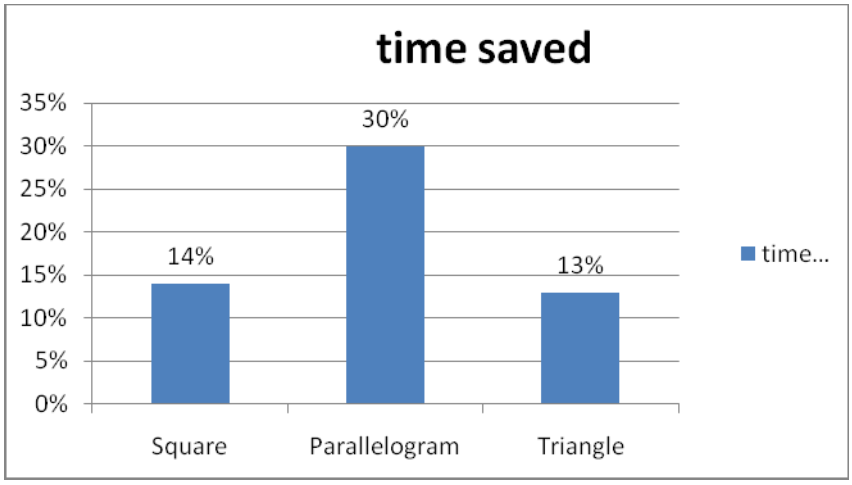

Figure 9 Percentage of Time Saved

6.2 Explain Trading Volume and Skills

The application above shows that how LFA has already benefited the math learning of U.S. children, but in addition LFA has already demonstrated its value for application to finance and economics.

A major strength of the U.S. economy is that it is built upon efficient and liquid financial markets. Thus, understanding the drivers for maintaining liquidity is of national interest. Average daily dollar values of U.S. equity traded in August 2007 was $14.14 billion, up approximately 126.2% from $6.25 billion in August 2006, according to

Knight Capital Group, Inc. This trading volume is the result of the important price discovery process that is required for successful allocation and investment decisions to be made by both "Wall and Main" streets. Protecting the future growth of trading volume in the regulated US financial centers is the major concern that led to the high-level meeting called by Hank Paulson. Unfortunately, Paulson's high-level meeting was unable to agree on the economic implications flowing from current regulatory trends.

From a theoretical perspective observed trading volumes are difficult to explain within the context of classical price taking (i.e., non strategic) economic theory. As a result, attempts to apply classical theory to draw implications from current regulatory trends in the US fall short. This identifies the importance for understanding trading volume by studying the strategic interaction between members of the trading crowd (e.g., market makers and market takers). An important component for understanding the strategic interaction between market makers and market takers is *how traders in the market learn to adjust* to each other as well as the economic conditions that impact financial markets in real time. Learning Factors Analysis (LFA) contributes to the understanding of how different members of the trading crowd learn and the implications of this for price discovery, efficiency and liquidity by providing a precise description of how, the strategic interaction between market makers and market takers with individual differences in abilities and prior information, results in the discovery of efficient prices and very large trading volumes.

6.2.1 The Market Data

The data under analysis is from the trading competition conducted at Carnegie Mellon University in xxx (year). The trader crowd consists of the MBA students and the MSCF students enrolled in the completion. The whole competition is comprised of four identical trials. Within each trial there are three different periods. Each period lasts for 300 seconds, standing for the first day of one of the consecutive years. In the first two periods, there are artificial news flown in and traders need constantly adjust their expectation upon the news. In the last trial of each period, there is no news and traders are supposed to form unified expectation. Thus the securities under trade had constant prices throughout last session.

For example, in the third period of trial one, there are two fixed-income securities under trade – security 1, a three-year coupon bond with face value of $100 and maturity in one year, and security 4, a zero-coupon bond with the same face value and maturity. As there is no news flown in, the interest rate is held constant and the market is meant to be highly efficient.

A rational trader is supposed to quote bid ask around the true constant prices all the time and only buy or sell if the market bid ask spread deviate from the true price. If every trader is rational, no trades will happen because if every trader will be quoting around true price, the bid ask spread will envelope the true price and no one is willing to buy or sell. However, we observed wide price fluctuations and huge trading volumes going on with the two securities in the last session. The research question is how do we explain this trading volume and the price discovery process.

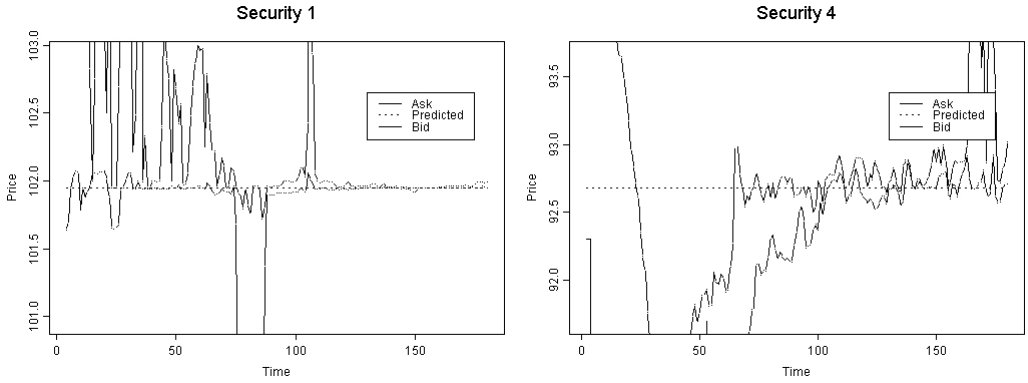

Figure 10 Prices of two securities

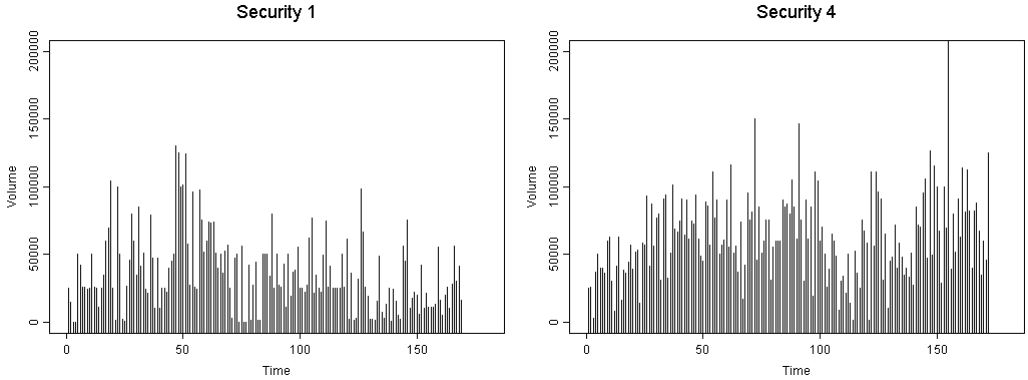

Figure 11 Trading volumes of the two securities

6.2.2 Applying LFA

The first crucial step in measuring traders' learning is to define the skills a trader may use and the situations where the skills are considered to be used correctly. For each security in the trading competition, we define two skills to describe the marketing making strategy – bid and ask -- and two skills to describe the market taking strategy – buy and sell.

Bid and ask stand for bidding to buy a security at a specific price and asking for selling a security at a specific price. The economics of market making would predict that a market maker should not bid above an asset's true value or ask below the true value. Thus, we define a correct execution (a not-wrong execution , more precisely) of bid to be bidding at a price no higher than the asset's true value, and ask to be asking at price no lower than the true value. Accordingly, we have two market taking skills defined in this study – Bid and Ask.

Buy and sell stand for buying a security at the best ask price and selling a security at its best bid price. According to the common notion of "Buy Low and Sell High", skill buy is defined to be correctly executed if a trader buys the security when the asset's value is above the best ask. Skill sell is defined to be correctly executed if a trader sells the security when the asset's value is below the best bid. Thus, we have four market taking skills defined in this study – Buy and Sell.

Then, for each security, we identified the market makers and market takers for it. Notice, the same trader can be both a market maker and a market taker for the same security.

With the skills and traders operationally defined, we applied the learning model to each security. Table 20 shows the parameter estimates for the skills.

Table 20 Parameter fits for the securities

	Security 1		Security 4	
	Skill (β)	Learning (γ)	Skill (β)	Learning (γ)
Ask	1.125	0.02	0.687	-0.003
Bid	0.592	0.09	2.9	-0.053
Buy	-0.965	-0.008	1.111	-0.03
Sell	-0.653	-0.044	-0.298	-0.026

The skill column shows the initial performance for market markers and market takers. For security 1, market markers performed better than market takers initially and over time. Buying for market takers was worse than selling.

The learning column shows their performance improvement overtime. For security 1, market makers performed better over time while market takers were making more mistakes.

For security 1, market makers excelled in bids and market takers were worse in sells

For security 4, market makers were best in bidding and market takers were worse in selling. Overtime, both groups make more mistakes.

Figure 12 shows the learning curves for each security. Figure 10 shows the prices of two securities over time.

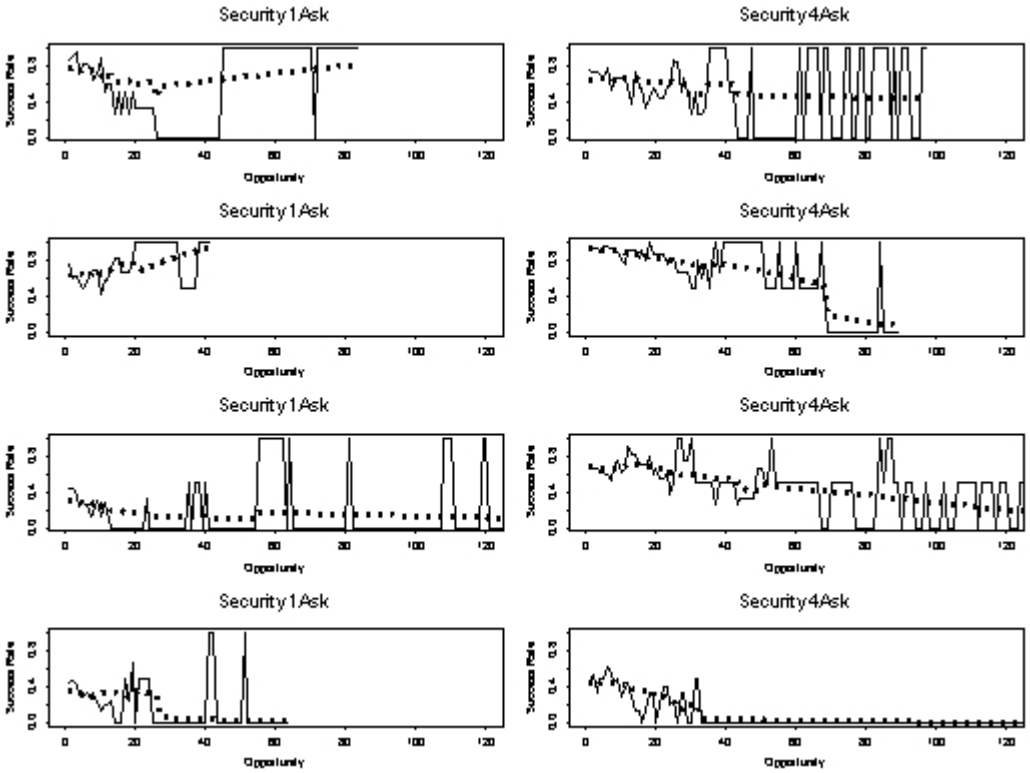

Figure 12 Learning curves in trial 1 period 3. The solid lines are the actual success rates of using each skill across the ordered number of practices. The dotted lines are the success rates predicted by the LFA model.

Another component of the trader learning model is the trader parameters. We break down the trader parameters by their market roles and the securities and plot each group's density. For both securities, the market takers show a two-modal distribution while the market takers is one modal. For security 1, the market takers and market makers are comparable in their prior knowledge. For security 4, market takers are better than market makers in their prior knowledge.

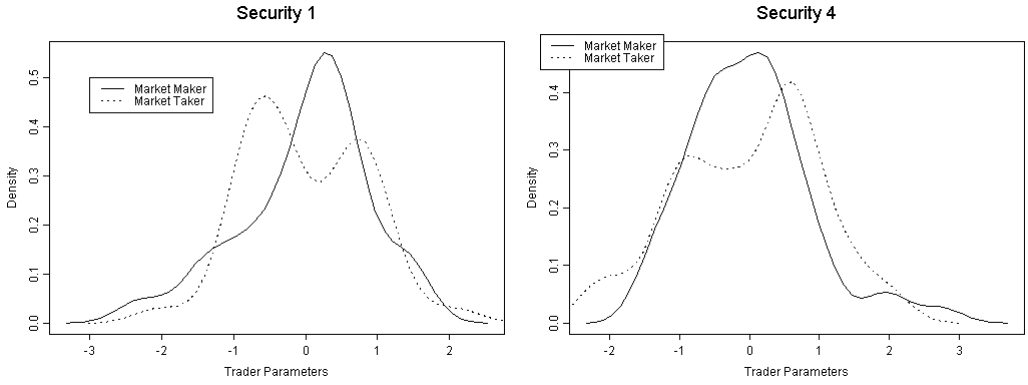

Figure 13 Density plots for the trader parameters of market makers vs. market takers

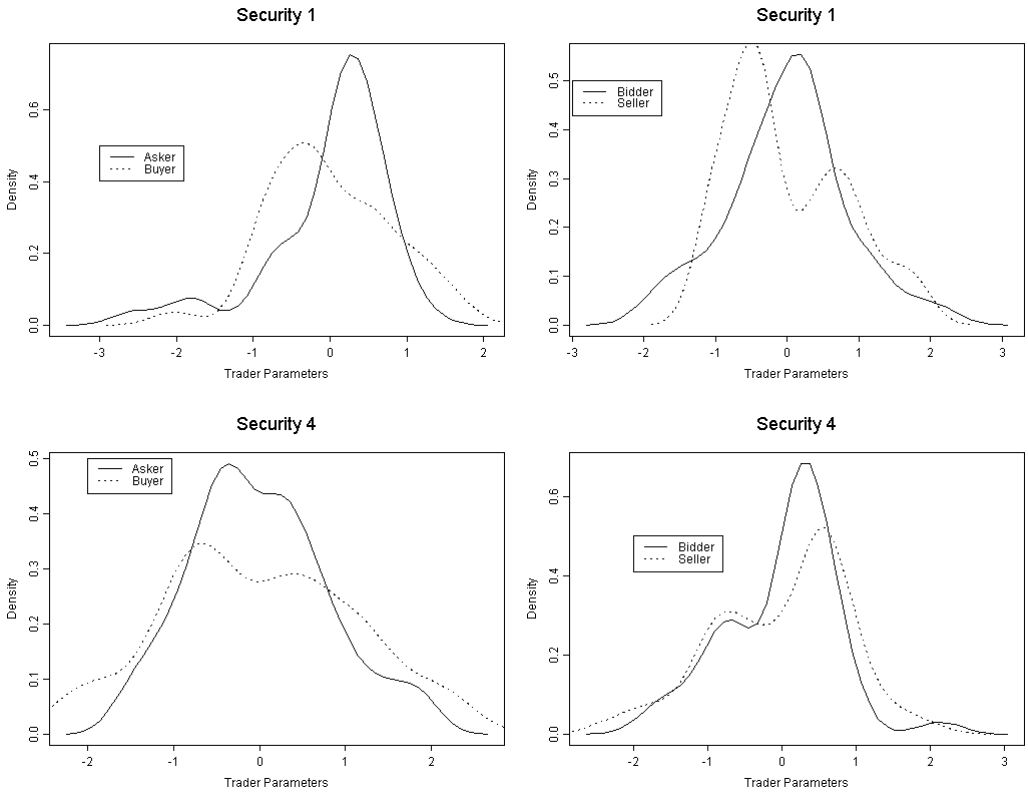

Figure 14 Density plots for the trader parameters of askers vs. buyers, bidders vs. sellers

In the simple setting studied a trade implies the market maker gains and the market taker loses, vice versa, or each is indifferent to trading. As a result, either no trades are predicted or no systematic behavior is predicted from indifference. However, our results reveal the unusual finding that the systematic behavior that resulted in efficient price discovery can be described by "negative learning." That is, contrary to traditional learning where repeated practice improves performance we observe the opposite, yet in the simple financial markets studied somewhat paradoxically this resulted in relatively efficient prices. We identify three fundamental drivers that underlie this result. First, the dynamically adjusting size of the bid/ask spread which tends to shrink over a trading period under the rules of the double auction institution creates an accuracy versus liquidity tradeoff for a market maker who is posting both bids and asks in an attempt to earn the spread from available liquidity. Second in a boundedly rational trading crowd there are individual differences in both abilities and the prior information brought to a trading period, and the double auction market institution provides an efficient matching mechanism for offsetting individual differences between market makers and market takers. Third, we detect that negative learning is associated with the presence of the winners curse phenomena. Combined these drivers which demonstrate how the market mechanism and a boundedly rational trading crowd interact to generate liquidity yet result in efficient price discovery.

7. Automatic Discovery of Q Matrices

Like the secrete key to the gate of the treasury, the Q matrix serves as key to predict student performance on questions. LFA heavily depends on the existence of such a matrix. The origination of such matrices involves extensive human expertise. Automatic discover of Q matrices may significantly reduce the human labor work in labeling Q matrices. Here I discuss two methods we experimented with. Each has its pros and cons.

7.1 Partial Ordering Knowledge Structure (POKS)

7.1.1 Motivation and the Data Set

This is a large dataset with over 16 million problems step performances provided by Carnegie Learning from the Bridge to Algebra Cognitive Tutor for the 2006-2007 school years. This tutor works by providing a systematic coverage with 44 units of pre-algebra content each of with has several sections. These sections consist of a problem type, which is composed of several steps or "item-types" which repeat over a sequence of similar problems. Students are not typically constrained in the order with which they do the steps within each problem. We want to determine a cognitive model that we could subsequently use to produce improvements to a tutor. While the current tutor had a skill model, we felt that it would be wise to question the validity of this model since it had been derived from human coding of item-types and therefore seemed subject to the possibility of human error and bias.

7.1.2 Introduction to POKS

POKS are one way to characterize the theory of knowledge spaces. Knowledge spaces describe how the learn units (or item-types in this paper) in a domain are learned in a constrained order [30]. Knowledge spaces have been investigated by many researchers using different methods [31, 32].

To explain the POKS method, first, we introduces the notations following [33]:

$A, B, \ldots$ the upper case Roman letters denote item-types in test

$A \Rightarrow B$ Knowing how to solve item-type A correctly leads to solving item-type B correctly (we have reversed this arrow notation in Figure 1.)

$P(B \mid A)$ the probability of getting item-type B right, given A was right

$P(\neg A \mid \neg B)$ the probability of getting item-type A wrong, given B was wrong

p_c the minimum probability that $P(B \mid A)$ and $P(\neg A \mid \neg B)$ need to hold

α_c the error of the POKS tests, which may be set differently for different tests

$N_{A \wedge B}$ the number of times that students get A right and B right*

$N_{A \wedge \neg B}$ the number of times that students get A right and B wrong*

$N_{\neg A \wedge B}$ the number of times that students get A wrong and B right*

$N_{\neg A \wedge \neg B}$ the number of times that students get A wrong and B wrong*

Because the independence of observations assumption of the statistical tests was strained when considering repetitions of the same item-type, these values were normalized by dividing the each by the total so that they summed to 1. The statistical tests then assumed a number of degrees of freedom equal to the number of subjects in

each pairwise comparison. This correction is overly conservative, but provides an unbiased correction for the sometimes grate between-subjects variability in the *N* of repetitions.

$CDFBinomial(x, n, p)$ the cumulative density function of a binomial distribution of n trials and p success probability

The idea of POKS is that if $A \Rightarrow B$ perfectly, we would expect $P(B \mid A) = 1$ and $P(\neg A \mid \neg B) = 1$. In reality, due to noises and imperfect $A \Rightarrow B$, we would expect the above two equalities not to hold exactly. Thus we can setup tests such that if $P(B \mid A)$ and $P(\neg A \mid \neg B)$ are above some threshold p_c, we can have some confidence of $A \Rightarrow B$.

There are three tests in total. The first two tests test whether $P(B \mid A)$ and $P(\neg A \mid \neg B)$ are above some threshold p_c, given the allowed test error α_c. The third test verify whether the conditional probability $P(B \mid A)$ and $P(\neg A \mid \neg B)$ are different from $P(B)$ and $P(\neg A)$

Test 1 returns true if $CDFBinomial(N_{A \wedge \neg B}, N_{A \wedge B} + N_{A \wedge \neg B}, 1 - p_c) < \alpha_c$

Test 2 returns true if $CDFBinomial(N_{A \wedge \neg B}, N_{\neg A \wedge \neg B} + N_{A \wedge \neg B}, 1 - p_c) < \alpha_c$

Test 3 returns true if the 2*2 contingency table of $N_{A \wedge B}$, $N_{A \wedge \neg B}$, $N_{\neg A \wedge B}$ and $N_{\neg A \wedge \neg B}$ passes a χ^2 test with error rate α_c

As we can see by examining the structure of the test, it relies on the covariance structure of the contingency table that is tabulated for each pair-wise item-type comparison. This covariance structure "places" each item-type in the POKS relative to the other item-types. By reflecting on this we can see that if we want to cluster the items based on the similarity of the required proficiencies, which would imply they require the same skills, we need a distance metric for item-types that a) captures that two items co-vary together and b) can cope with the fact that two items may not be equally difficult despite having the very similar covariance structures. Requirement *a* means we need a distance metric that captures the structure of the contingency tables for item-type X_1 as compared to the contingency tables for item-type X_2. Requirement *b* means that this metric probably should not capture the structure of the tables relative to the outcome of performance X_1 or X_2. Rather, we should describe a distance metric that is computed independently for those cases where X_1 or X_2 is a success or failure. Requirement b is important for the purpose here because the tutor introduces item-types in a fixed order. This fixed order means differences in average performance between item-types may be caused by learning. However, this difference in performance between item-types that represent the same skill should not greatly alter the contingencies given the response is a success or failure.

To do this comparison of the covariance structure it helps to consider the data for two item-types (X_1 and X_2 , which will be correlated) as being organized into two columns of contingency tables where the rows are all other possible Y item-types. (Each contingency table being organized with X frequency results for A and ~A as rows and Y frequency results for B and ~B as columns.) For each X_n by Y_n contingency table we will compute 2 values one for when X_n is a success and one for when X_n is a failure separately. In each of these cases, the B and ~B contingency table values are represented using a logit (log odds) to represent the strength of the odds B:~B on a continuous scale that captures the relationships and its relative strength better than the raw frequencies. Because these logits do not capture the effect of frequency of A or ~A and only capture the relative frequency of B vs. ~B they are not reactive to learning of A that does not affect the patterns of B vs. ~B, nor are they reactive to difference in the n of observations

of the B:~B results. Using this procedure we compute this logit for each contingency table row for each row item-type.

At this point we can describe vectors of logits for each column item-type X_1 and X_2 (getting values for A and ~A as logits for each item X and Y item type to compose the vectors) and compute their correlation to determine the nearness of two item-types in the knowledge space. To do this clustering we used a simple agglomerative hierarchical clustering to cluster item-types into clusters that represented a new grainsize which implies clustered items shared the same performance requirements (skill). This new method shares similarities with correlation clustering methods that have proven useful for graph partitioning [32] and is described further in the next section.

The analysis methods described above were run with a clustering coefficient of 0.55, a p_c value of 0.35 and a_c values of 0.15/ These parameters were selected to limit the number of links and restrict the clustering to obtain a graph that would provide us examples we could analyze (one link was also removed by the requirement for a >0 duration correlation between POKS linked item-types). This procedure follows Desmarais who has discussed how a_c and p_c need to be adjusted to the specific characteristics of the dataset [33]. Additionally, before analysis, we removed students with less than 250 transactions, and we removed item-types with less than 2000 transactions. These filters reduced our dataset to 1073 students with a total of 1,099,642 outcomes (the quantity of response latencies was lower since we only accepted latencies for trials with a successful outcome.) The analysis was restricted to this subset (which included all transactions from 3 contiguous units of the 44 in the tutor to make it manageable to show a graphic of the results. Larger numbers of units/item-types could not be made to fit on a single page.

7.1.3 Results from POKS

Figure 1 shows the POKS graph obtained from this analysis and corresponds to the groupings in Table 1 which provides additional statistics to help interpret the results. The ovals in Figure 1 represent collections (or individual) item-types which were a function of the clustering procedure (also grouped in Table1). Item-type Label indicates the following information (probabilitycorrect_Uniname_sectionnumber_SkillIDnumber). The table also provides the average duration and total number of database observations (Rps in 1000s) for each item-type . Colors indicate the majority unit membership for the grouped item-types, where LCM – least common multiple unit, GCF – Greatest common factor unit, and FracRep involves a visual and written fraction representation unit.

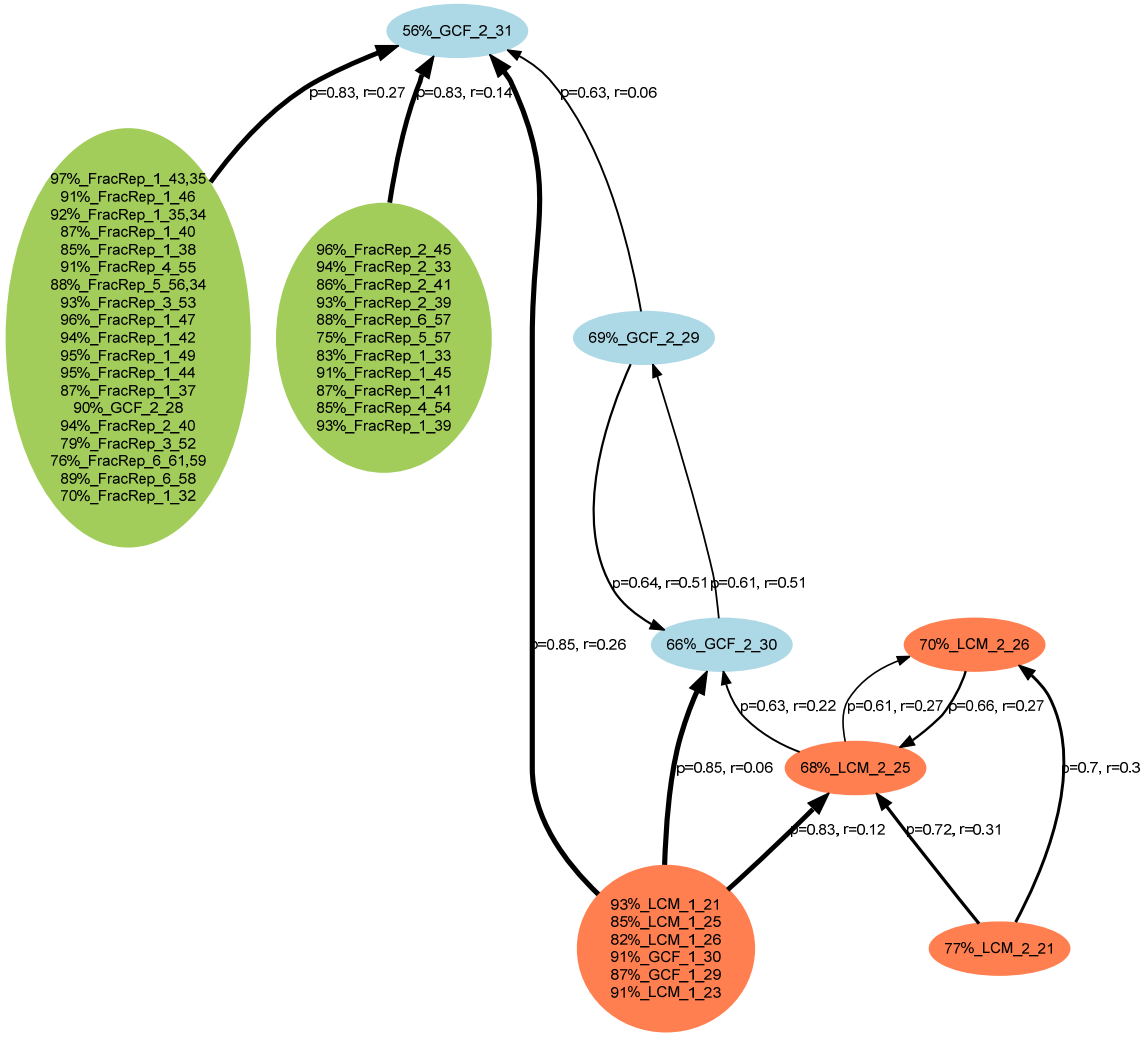

Figure 1. Graph structure described in the results section.

The clustering of these groups of item-types indicates that the pattern of contingency tables and the pattern of correlations were similar for these item-types such that if item-types A_1 and A_2 are in a cluster it indicates they have similar relationship to the other item-types in the analysis. By extension we can suppose that this similar place in the covariance structure suggests that performance for these item-types is constrained by the same knowledge component. The fact that this clustering occurs suggests that the human coders used cognitively irrelevant features to code the item-types. For example consider the green ovals in Figure 1. The right oval includes a variety of item-types that might be described as understanding the denominator, while the left oval includes item-types that deal with the numerator. (We can see that much of this clustering appeared necessary because the human coder assumed that the shape being used in the example required a different skill depending on whether it was a vertical bar, horizontal bar, circle, square, or number line. In contrast, the clustering method lumped these skills together indicating they are actually the same proficiency.) By splitting these groups into

separate skills the human coder delinked these proficiencies relative to the tutor's automatic scheduling mechanisms. So, for example, if a student does very well on these clustered item-types as they are introduced it will not result in less practice for the other items that our analysis suggests are in the cluster. Therefore by proposing this cluster we can address learning of the concept more efficiently because we can model transfer between item-types that are controlled by the same underlying proficiencies. Modeling transfer between item-types allows us to know when a particular concept, skill or procedure has been mastered despite the fact that we may not have given a student examples of all the item-types in the cluster. Item types with the same ID appeared in different units, but were labeled with the same skill by the human coders. As we can see in Table 1, our clustering method tended to confirm these preexisting coding despite indicating further clustering was possible.

7.2 Exponential-Family Principle Component Analysis (EPCA)

7.2.1 Principle Component Analysis (PCA)

Principle Component Analysis (PCA) is a popular machine learning method for feature extraction and dimension reduction. It is traditionally viewed as either the maximization of the variance of the data projected on a lower dimension space. From a generative point of view, PCA can be expressed as finding the mapping from data space to latent space with a lower dimension than the data space. Specifically, the observed data are generated by a linear transformation of the latent variables plus Gaussian noise [34].

7.2.2 Exponential-Family Principle Component Analysis (EPCA)

EPCA, a generalization of PCA, addresses the problem when the noises are not Gaussian. The general idea of EPCA is that it views each data point $X_{ij} \in \mathbb{R}^d$ as the realization of an exponential family random variable with natural parameter θ_{ij}, which belongs to a lower dimensional subspace. A link function $g(.)$ is used to connect X_{ij} to θ_{ij} [35].

$$X_{ij} \sim P(X_{ij} \mid \theta_{ij})$$
$$\theta_{ij} = U_{i.} * V_{.j}$$
$$\Theta = UV$$

Where

X_{ij} the observed value in the data matrix X

θ_{ij}, the parameter of the latent random variable that generates X_{ij}

U, V, the factored matrix

$U_{i.}, V_{.j}$, the ith row of U, the jth column of V

It finds θ_{ij} by minimizing the loss function of U and V with respect to the data matrix X. The loss function is

$$Loss(U,V) = -\log p(X;U,V) = -\sum_{i,j} p(X_{ij} \mid \theta_{ij})$$

An efficient method designed by Gordon [13] and Singh [36]estimates Θ by alternatively optimization the loss function while holding one of the U,V matrices constant one at a time.

7.2.3 Application of EPCA for Automatic Discovery of Q Matrices

The typical data for student learning are student responses on test items. The responses are usually 1s or 0s, corresponding to correct or failure on the items. By crosstabbing the student performance data on students and questions, we get so called student-item matrices, which can be thought as generated from binomial distribution, a candidate for EPCA. We proposed four formulations of EPCA with increasing complexities to answer various research questions. For each formulation, we show the factored matrices as well as the optimization form. The advantages and disadvantages of each formulation are discussed.

$$X_{ij} \sim Bernoulli(\theta_{ij})$$
$$E[X_{ij}] = \text{logit}[\theta_{ij}]$$

7.2.3.1 Formulation 1 -- directly apply Bernoulli EPCA to student-item matrices (baseline)

$$\begin{pmatrix} \beta_{11} & \beta_{12} & \beta_{13} \\ \beta_{21} & \beta_{22} & \beta_{23} \\ \beta_{31} & \beta_{32} & \beta_{33} \\ \beta_{41} & \beta_{42} & \beta_{43} \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} q_{11} & q_{21} & q_{31} & \cdots \\ q_{12} & q_{22} & q_{32} & \cdots \\ q_{13} & q_{23} & q_{31} & \cdots \end{pmatrix}$$
$$\beta_{ij,} q_{ij} \in \mathrm{R}$$

$$\min Loss(U,V;X)$$
$$Loss(U,V;X) = \sum_{ij} X_{ij} \ln(U_{i.}{}^{T} V_{.j}) + (1 - X_{ij}) \ln(1 - U_{i.}{}^{T} V_{.j})$$

This is a direct translation of the model to the problem. With the existing implementation of EPCA, using it is not hard. However, due to the large number of elements of in U and V, the estimation errors could be large. In addition, it is hard to interpret the meanings of entries.

7.2.3.2 Formulation 2 -- directly apply Bernoulli EPCA to student-item matrices and add student parameter and one 1s row to the skill-step matrix

$$\begin{pmatrix} \theta_1 & \beta_{11} & \beta_{12} & \beta_{13} \\ \theta_2 & \beta_{21} & \beta_{22} & \beta_{23} \\ \theta_3 & \beta_{31} & \beta_{32} & \beta_{33} \\ \theta_4 & \beta_{41} & \beta_{42} & \beta_{43} \\ & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ q_{11} & q_{21} & q_{31} & \cdots \\ q_{12} & q_{22} & q_{32} & \cdots \\ q_{13} & q_{23} & q_{31} & \cdots \end{pmatrix}$$

$$\theta_i, \beta_{ij,} q_{ij} \in \mathrm{R}$$

$$\min Loss(U, V; X)$$

This formulation accounts for student proficiency. However, it is still hard to interpret the meanings of entries.

7.2.3.3 Formulation 3 -- directly apply Bernoulli EPCA to student-item matrices, add student parameter and one 1s row to the skill-step matrix, and constrain beta and q

$$\begin{pmatrix} \theta_1 & \beta_{11} & \beta_{12} & \beta_{13} \\ \theta_2 & \beta_{21} & \beta_{22} & \beta_{23} \\ \theta_3 & \beta_{31} & \beta_{32} & \beta_{33} \\ \theta_4 & \beta_{41} & \beta_{42} & \beta_{43} \\ & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ q_{11} & q_{21} & q_{31} & \cdots \\ q_{12} & q_{22} & q_{32} & \cdots \\ q_{13} & q_{23} & q_{31} & \cdots \end{pmatrix}$$

$$\theta_i \in \mathrm{R}$$

$$\beta_{ij} \le 0$$

$$q_{ij} \ge 0$$

$$\min Loss(U, V; X)$$

$$\beta_{ij} \le 0$$

$$q_{ij} \ge 0$$

In this formulation, matrix V starts to behave like a Q matrix and beta has a similar meaning to skill difficulty. However, the implementation is significantly harder.

7.2.3.4 Formulation 4 -- directly apply Bernoulli EPCA to student-item matrices, Add student parameter and one 1s row to the skill-step matrix, and Constrain beta and q more

$$\begin{pmatrix} \theta_1 & \beta_{11} & \beta_{12} & \beta_{13} \\ \theta_2 & \beta_{21} & \beta_{22} & \beta_{23} \\ \theta_3 & \beta_{31} & \beta_{32} & \beta_{33} \\ \theta_4 & \beta_{41} & \beta_{42} & \beta_{43} \\ & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ q_{11} & q_{21} & q_{31} & \cdots \\ q_{12} & q_{22} & q_{32} & \cdots \\ q_{13} & q_{23} & q_{31} & \cdots \end{pmatrix}$$

$$\theta_i \in \mathrm{R}$$

$$\beta_{ij} \le 0$$

$$0 \le q_{ij} \le 1$$

$$\min Loss(U,V;X) + c\sum_{ij} \beta_{ij}^{\ 2}$$

$$\beta_{ij} \le 0$$

$$0 \le q_{ij} \le 1$$

V in this formulation is even closer to a Q matrix. The implementation difficulty is similar to that of formulation 3.

7.2.4 Complications of applying EPCA to EDM

Often times students are given different test items. A value of zero in the student-item matrix may mean either that the student failed on this item or that the student may not have done the items. A straight forward application of EPCA to the student item matrix may leads to erroneous results. One solution to that is to add a weight matrix for the student-item matrix. If the student has done the item, the weight is 1. Otherwise the weight is 0.

$$\min Loss(U,V;X)$$

$$Loss(U,V;X) = \sum_{ij} W_{ij}(X_{ij} \ln(U_{i.}^{\ T} V_{.j}) + (1 - X_{ij}) \ln(1 - U_{i.}^{\ T} V_{.j}))$$

7.2.5 Evaluation of EPCA

A nice feature of LFA is that all the skill labels in a Q matrix have interpretable meanings. Although EPCA is able to return a Q matrix approximation from the data, it is unable to label the columns on the Q matrix. Instead of evaluating the interpretability of the Q matrices found by EPCA, we focus on the prediction ability of EPCA. The following illustrates the Fold-in algorithm that PCA uses to predict student performance, given new items or new students.

Step 1. Factor the existing matrix using EPCA, using training data

$$[U_{train}, V_{train}] = EPCA(X_{train})$$

$$X_{train} = \begin{matrix} st1 \\ st2 \\ st3 \\ st4 \\ st5 \end{matrix} \begin{pmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \\ X_{41} & X_{42} & X_{43} \\ X_{51} & X_{52} & X_{53} \end{pmatrix}$$

$$U_{train} = \begin{pmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \\ \beta_{31} & \beta_{32} \\ \beta_{41} & \beta_{42} \\ \beta_{51} & \beta_{52} \end{pmatrix}$$

$$V_{train} = \begin{pmatrix} q_{11} & q_{21} & q_{31} & q_{41} & q_{51} \\ q_{12} & q_{22} & q_{32} & q_{42} & q_{52} \end{pmatrix}$$

Step 2. Use a smaller number of students to estimate the new V column v_fold, holding U fixed, given the new item; using logistic regression to estimate v_fold

$$v_{fold} = \text{logisticRegression}(X_{fold}, U_{train\_fold})$$

$$X_{fold} = \begin{matrix} st1 \\ st2 \end{matrix} \begin{pmatrix} X_{15} \\ X_{25} \end{pmatrix}$$

$$U_{train\_fold} = \begin{pmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{pmatrix}$$

$$v_{fold} = \begin{pmatrix} q_{51} \\ q_{52} \end{pmatrix}$$

Step 3. Use the new V column and old U to get student performance on new item

$$X_{test} = U_{train\_test} v_{fold}$$

$$U_{train\_test} = \begin{pmatrix} \beta_{31} & \beta_{32} \\ \beta_{41} & \beta_{42} \\ \beta_{51} & \beta_{52} \end{pmatrix}$$

$$v_{fold} = \begin{pmatrix} q_{51} \\ q_{52} \end{pmatrix}$$

Then we can compare the actual student performance vs. the predicted student performance using

$$error = \frac{1}{n} \sum_{i=1}^{n} (Actual_i - Prediction_i)^2$$

7.2.6 Results

7.2.6.1 Simulated Data

We simulated data from AFM with 100 students, 3 skill, 9 items. Every student does all the 9 items. The student parameter is taken from a normal distribution with 0 mean and 1 standard deviation. The three skills have difficulty parameter values as -2.2,0,2.2. The true Q matrix is

Time|Skill	A	B	C
T1_100	1	0	0
T2_010	0	1	0
T3_001	0	0	1
T4_100	1	0	0
T5_010	0	1	0
T6_001	0	0	1
T7_100	1	0	0
T8_010	0	1	0
T9_001	0	0	1

Given the existing implementation of EAPS, we used the first formulation described above. To get a more comprehensive view on the performance of EPCA, we designed seven experiments.

Experiment 1 (EPCA-true U, true V) -- we construct an (approximately) true U matrix and an (approximately) true V matrix to compute the fitting error and the prediction error directly without using the EPCA algorithm. We construct the U matrix from adding each skill parameter β to the corresponding student parameter α, and construct the V matrix by taking every element from the given Q matrix. The word "approximate" comes from the fact that even though the U, V matrices we construct contain elements of truth, they may not be best one for EPCA.

$$U_{true} = \begin{pmatrix} \alpha_1+\beta_1 & \alpha_1+\beta_2 & \alpha_1+\beta_3 & \cdots \\ \alpha_2+\beta_1 & \alpha_2+\beta_2 & \alpha_2+\beta_3 & \cdots \\ \alpha_3+\beta_1 & \alpha_2+\beta_2 & \alpha_2+\beta_3 & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

$$V_{true} = Q$$

Experiment 2 (EPCA-true U, unknown V) -- we pass the student-item matrix and the true U matrix to EPCA and have EPCA get the V matrix. The idea here is to hold U fixed in EPCA while getting V.

Experiment 3 (EPCA- unknown U, unknown V) -- we pass only the student-item matrix to EPCA and let EPCA get both U, V matrices.

Experiment 4 (Random prediction) – we draw a random prediction from a uniform distribution between 0 and 1. The random prediction is meant to serve as benchmark of being guessing at random.

Experiment 5 (AFM) – we use AFM and the true V matrix to do the prediction.

Experiment 6 (PAFM) – we use PAFM and the true V matrix to do the prediction. The two LFA models are included here not as a mean to be a fair comparison for LFA and EPCA on this simulated data set because the data is generated from AFM. They are listed as another benchmark on how low cross validation error could be. The learning term is omitted from the both models.

Experiment 7 (EPCA – U_PAFM, unknown V) – we use the U matrix estimated from PAFM and pass it to EPCA. Then EPCA estimates the V matrix. The goal is to combine the advantages of PAFM and EPCA and have PFAM generate alternative V matrices.

For each experiment, we use 9 fold cross validation on the data set. Each test fold has one single item done by 100 students.

As shown in the table below, some interesting observations can be seens

1) The training error steadily decreases from experiment 1 to experiment 3 while the cross validation error steadily increases. This is a sign that EPCA overfits the data more and more, as EPCA gets more parameters to changes from experiment 1 to experiment 3, although the existing EPCA implementation includes a regularization term to reduce over fitting.
2) AFM and PAFM have the lowest cross validation errors. It is not surprising since the data were generated from AFM. PAFM has a higher training error but a lower cross validation error than AFM. showing the penalization helps with the out-of-sample prediction. The cross validation error for AFM and experiment 1 are close, hinting that AFM and EPCA with true U and true V are similar in structure.
3) With the help from PAFM, in experiment 7 the cross validation error is lower in EPCA, suggesting that the information from PAFM may help EPCA to generate the V matrices better than without.

Table 21 Average training error and cross validation error in each of the three experiments

	Training Error	Cross Validation Error
Experiment 1 (true U, true V)	0.174	0.166
Experiment 2 (EPCA-true U, unknown V)	0.123	0.197
Experiment 3 (EPCA- unknown U, unknown V)	0.030	0.228
Experiment 3 (EPCA- unknown U, unknown V, regularization parameter =.01)	0.038	0.183
Experiment 3 (EPCA- unknown U, unknown V, regularization parameter =.1)	0.050	0.172

Experiment 3 (EPCA- unknown U, unknown V, regularization parameter =.5)	0.067	0.173
Experiment 4 (Random prediction)	0.339	0.329
Experiment 5 (AFM)	0.114	.161
Experiment 6 (PAFM)	0.124	0.149
Experiment 7 (EPCA – U_PAFM, unknown V)	0.113	0.20
Experiment 7 (EPCA – U_PAFM, unknown V, regularization parameter =.1)	0.1119948	0.2032629

	Training Error	Cross Validation Error
Experiment 3 (EPCA- unknown U, unknown V, regularization parameter =.5, numskills = 1)	0.1798677	0.2319537
Experiment 3 (EPCA- unknown U, unknown V, regularization parameter =.5, numskills = 2)	0.1004001	0.1687146
Experiment 3 (EPCA- unknown U, unknown V, regularization parameter =.5, numskills = 3)	0.067	0.173
Experiment 3 (EPCA- unknown U, unknown V, regularization parameter =.5, numskills = 4)	0.03105644	0.1835261
Experiment 3 (EPCA- unknown U, unknown V, regularization parameter =.5, numskills = 5)	0.01465372	0.1862017
Experiment 3 (EPCA- unknown U, unknown V, regularization parameter =.5, numskills = 6)	0.006558789	0.2009614

The following two tables show two Q matrices from two cross validation tests on the same data set from Experiment 7. In Cross Validation on item 1, the item 1 portion of the matrix is not estimated as item 1 was omitted in the construction of V in EPCA. The item 2 portion of the matrix is not estimated as item 2 was omitted in Cross Validation on item 2. The V matrixes got from EPCA vary from one to another in each validation set from each test.

Table 22 Two Q matrices from EPCA on the same data

	Cross Validation on item 1			Cross Validation on item 2		
	A	B	C	A	B	C
T1_100				0.47	0.06	0.62
T2_010	0.55	0.86	0.62			
T3_001	0.32	0.81	0.10	0.19	0.88	0.43
T4_100	0.08	-0.46	0.24	0.39	-0.09	0.43
T5_010	0.91	0.56	0.73	0.96	0.17	0.55
T6_001	0.28	0.90	0.21	0.05	0.53	-0.02
T7_100	0.44	0.36	0.02	0.45	0.14	0.56
T8_010	0.68	0.12	1.21	0.72	0.93	0.34
T9_001	0.23	0.51	0.26	0.08	1.36	0.06

7.2.6.2 EPCA vs. LFA on EAPS Data

A more fair comparison of EPCA and LFA would be on a real data set when the true Q matrix is unknown. By a similar methodology in experiment 3 for the simulated data, we did leave-one-out cross validation using EPCA and LFA on the EAPS data. AFM requires the knowledge of the Q matrix. Giving a Q matrix to AFM gives AFM an advantage in the comparison. Thus, we used a Q matrix searched by AFM from a P matrix. The table below shows the Q matrix discovered by LFA.

Item	numCategory-hard-bball	numDifficulty-hard	origArith-div	presentation-equation	unknownPosition-result	unknownPosition-start
bball-equation-result-easy-div	0	0	1	1	1	0
bball-equation-result-easy-mult	0	0	0	1	1	0
bball-equation-result-hard-div	1	1	1	1	1	0
bball-equation-result-hard-mult	1	1	0	1	1	0
bball-equation-start-easy-div	0	0	0	1	0	1
bball-equation-start-easy-mult	0	0	1	1	0	1

We chose three experiments for the EAPS data.

Experiment 1 (EPCA- unknown U, unknown V) -- we pass only the student-item matrix to EPCA and let EPCA get both U, V matrices. We set the number of skills to be 3.

Experiment 2 (PAFM) – we use PAFM and the LFA discovered V matrix to do the prediction.

Experiment 3 (EPCA – U_PAFM, unknown V) – we use the U matrix estimated from PAFM and pass it to EPCA. Then EPCA estimates the V matrix. Here the number of skills in EPCA is 6, determined from the number of skills in the V matrix for PAFM.

To use EPCA on this data, we encountered the issue of data sparcity – namely, most of the cells in the student-item matrix being zero indicate the lack of the observation rather than student making mistakes on the items. By applying the weight matrix indicated in section 7.2.4, we are able to get the U, V matrices. We also apply the weight matrix to the folding procedure.

As shown in the table below, some interesting observations can be seen:

1) EPCA still suffers from overfitting the training data, demonstrated by experiment
2) PAFM confirmed best.
3) With U matrix from PAFM does not help with EPCA at all in experiment 3. This may be caused by the sparcity in the student-item matrix and there are simply not enough data to estimate the large number of parameters well.

Table 23 Cross Validation Errors by EPCA and LFA on the EAPS data

	Training Error	Cross Validation Error
Experiment 1 (EPCA- unknown U, unknown V)	0.004	0.272
Experiment 2 (PAFM)	0.129	0.17
Experiment 3 (EPCA – U_PAFM, unknown V)	0.35	0.39

7.2.6.3 What caused over fitting in EPCA?

One explanation for the over fitting in EPCA is that there are too many parameters for EPCA to fit in formulation 1. The number of parameters in AFM is the number of students plus the number of skills (times 2 if there are learning components). EPCA formulation 1 has the number of parameters as the number of students times the number of skills plus the number of skills times the number of items. Take the simulated data as the example, AFM has 100 + 3 = 103 parameters while EPCA formulation 1 has 100*3 + 3*9 = 327 parameters. The total number of data points in the simulated data is 900. Clearly, the overly large number of parameters with respect to the data available leads to over fitting.

To reduce the number of parameters in EPCA, one alternative to formulation 1 can be as follows

Formulation 1 alternative -- directly apply Bernoulli EPCA to student-item matrices with equality constraints

$$\begin{pmatrix} \beta_1 & \beta_1 & \beta_1 \\ \beta_2 & \beta_2 & \beta_2 \\ \beta_3 & \beta_3 & \beta_3 \\ \beta_4 & \beta_4 & \beta_4 \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} q_{11} & q_{21} & q_{31} & \cdots \\ q_{12} & q_{22} & q_{32} & \cdots \\ q_{13} & q_{23} & q_{31} & \cdots \end{pmatrix}$$

$$\beta_{1j} = \beta_{2j} = \beta_{3j} = \beta_j, \forall j$$

$$q_{ij} \in \mathrm{R}$$

With all student-item parameters constrained to be equal within each student, EPCA needs to estimate as the same number of parameters as AFM, which may boost EPCA's

performance. Experiment 2 in the simulated data has this flavor by giving EPCA a student-item matrix and requires EPCA only to estimate the skill-item matrix. The performance of EPCA on this experiment was higher than in experiment 3.

8. Conclusions and Future Work

9. Bibliography

1. Koedinger, K.R. and J.R. Anderson, *Intelligent Tutoring Goes To School in the Big City.* International Journal of Artificial Intelligence in Education, 1997(8): p. 30-43.
2. Koedinger, K.R., et al., *Carnegie Learning's Cognitive Tutor™: Summary Research Results*. 2002, Available from Carnegie Learning, Inc., http://www.carnegielearning.com/approach_research_reports.cfm.
3. Sarkis, H., *Cognitive Tutor Algebra 1 Program Evaluation*. 2004, Available from Carnegie Learning, Inc., 1200 Penn Avenue, Suite 150, Pittsburgh, PA 15222.
4. Nathan, M.J., et al. *Representational fluency in middle school: A classroom-based study.* in *the 24th Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education.* 2002.
5. Nathan, M.J. and A. Petrosino, *Expert Blind Spot Among Preservice Teachers.* American Educational Research Journal, 2003. **40**(4): p. 905–928.
6. Nathan, M.J., S.D. Long, and M.W. Alibali, *The symbol precedence view of mathematical development: A corpus analysis of the rhetorical structure of algebra textbooks.* Discourse Processes, 2002. **33**(1): p. 1–21.
7. Nathan, M.J., K.R. Koedinger, and M.W. Alibali. *Expert blind spot: When content knowledge eclipses pedagogical content knowledge*. in *the Third International Conference on Cognitive Science*. 2001. Beijing, China: University of Science and Technology of China Press.
8. Nathan, M.J. and K.R. Koedinger, *An investigation of teachers' beliefs of students' algebra development.* Cognition and Instruction, 2003. **18**(2): p. 207–235.
9. Nathan, M.J. and K.R. Koedinger, *Teachers' and researchers' beliefs about the development of algebraic reasoning.* Journal for Research in Mathematics Education, 2000. **31**: p. 168–190.
10. Koedinger, K.R. and M.J. Nathan, *The real story behind story problems: Effects of representations on quantitative reasoning.* The Journal of the Learning Sciences, 2003.
11. Koedinger, K. and M.J. Nathan, *Teachers' notions of students' algebra problem-solving difficulties.*, in *the annual meeting of the James S. McDonnell Foundation Program for Cognitive Studies for Educational Practice*. 1997.
12. Russell, S. and P. Norvig, *Artificial Intelligence: A Modern Approach*. 2nd ed. 2003: Prentice Hall
13. Gordon, G., *Generalized2 Linear2 Models*, in *Annual Conference on Neural Information Processing Systems 2002*. 2002.
14. Barnes, T., *The Q-matrix Method: Mining Student Response Data for Knowledge*, in *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*. 2005.
15. Tatsuoka, K., *Rule space: An approach for dealing with misconceptions based on item response theory.* Journal of Educational Measurement, 1983. **20**(4): p. 345-354.

16. DiBello, L., W. Stout, and L. Roussos, *Unified cognitive/psychometric diagnostic assessment likelihood-based classification techniques*, in *Cognitively diagnostic assessment*, P. Nichols, S. Chipman, and R. Brennan, Editors. 1995, Erlbaum: Hillsdale , NJ. p. 361-389.
17. Embretson, S., *Multicomponent Response Models*, in *Handbook of Modern Item Response Theory* W.V.D. Linden and R.K. Hambleton, Editors. 1997, Springer.
18. von Davier, M., *A General Diagnostic Model Applied to Language Testing Data*. 2005, Educational Testing Service.
19. Corbett, A.T. and J.R. Anderson, *Knowledge tracing: Modeling the acquisition of procedural knowledge*, in *User Modeling and User-Adapted Interaction*. 1995. p. 253-278.
20. Newell, A. and P. Rosenbloom, *Mechanisms of Skill Acquisition and the Law of Practice*, in *Cognitive Skills and Their Acquisition*, J. Anderson, Editor. 1981, Erlbaum Hillsdale NJ
21. Nichols, P., S. Chipman, and R. Brennan, *Cognitively diagnostic assessment*. 1995, Hillsdale , NJ: Erlbaum.
22. Davier, M.v., *A General Diagnostic Model Applied to Language Testing Data*. 2005, Educational Testing Service.
23. Fischer, G.H., *Linear logistic test models: Theory and application*, in *Structural Models of Thinking and Learning*, H. Spada and W.F. Kempf, Editors. 1977, Bern: Huber. p. 203-25.
24. Junker, B. and K. Sijtsma, *Cognitive Assessment Models with Few Assumptions and Connections with Nonparametric Item Response Theory.* Applied Psychological Measurement, 2001. **25**: p. 258-272.
25. Ur, S. and K. VanLehn, *STEPS: A Simulated, Tutorable Physics Student.* Journal of Artificial Intelligence in Education, 1995. **6**(4): p. 405-437.
26. Baffes, P. and R.J. Mooney, *A Novel Application of Theory Refinement to Student Modeling*, in *Thirteenth National Conference on Artificial Intelligence*. 1996: Portland OR
27. Harrell, F.E., *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. 2001: Springer.
28. Wasserman, L., *All of Statistics: A Concise Course in Statistical Inference*. 2004: Springer
29. Koedinger, K.R. and B.A. MacLaren, *Developing a Pedagogical Domain Theory of Early Algebra Problem Solving*. 2002, Human Computer Interaction Institute, Carnegie Mellon University.
30. Falmagne, J.-C., et al., *Introduction to knowledge spaces: How to build, test, and search them.* Psychological Review, 1990. **97**(2): p. 201-224.
31. Falmagne, J.-C., et al., *The assessment of knowledge in theory and in practice.* Institute for Mathematical Behavioral Sciences, 2003. **Paper 26**.
32. Desmarais, M.C., Gagnon, M., *Bayesian student models based on item to item knowledge structures*, in *First European Conference on Technology Enhanced Learning*. 2006: Crete, Greece.
33. Desmarais, M.C., Maluf, A., Liu, J., *User-expertise modeling with empirically derived probabilistic implication networks.* User Modeling and User-Adapted Interaction, 1996. **5**(3-4): p. 283-315.

34. Tipping, M.E. and C.M. Bishop, *Probabilistic Principal Component Analysis*. Journal of the Royal Statistical Society, 1999. **61**: p. 611–622.
35. Collins, M., S. Dasgupta, and R.E. Shcapire, *A Generalization of Principal Component Analysis to the Exponential Family*. Annual Conference on Neural Information Processing Systems 2001, 2001.
36. Singh, A.P. and G.J. Gordon. *A Unified View of Matrix Factorization Models*. in *Machine Learning and Knowledge Discovery in Databases, European Conference (ECML/PKDD)* 2008.