Generalized Learning Factors Analysis: Improving cognitive Models with Machine Learning

Hao Cen

April 2009
CMU-ML-09-102

MACHINE LEARNING
DEPARTMENT

Carnegie Mellon.

Generalized Learning Factors Analysis: Improving Cognitive Models with Machine Learning

Hao Cen

April 2008
CMU-ML-09-102

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania

Thesis Committee:
Kenneth Koedinger, Chair
Brian Junker
Geoff Gordon
Noel Walkington, Mathematical Sciences

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by the Department of Energy under contract no. R305B070487, the Department of Education under contract no. R305K030140, the Department of Energy (WPI) under contract no. 0821622001, and the National Science Foundation (PSLC) under contract no. SBE-0354420. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: cognitive models, intelligent tutoring systems, machine learning, educational data mining, learning factors, psychometrics, additive factor models, latent variable models, exponential principal component analysis, logistic regression, combinatorial search

To my parents

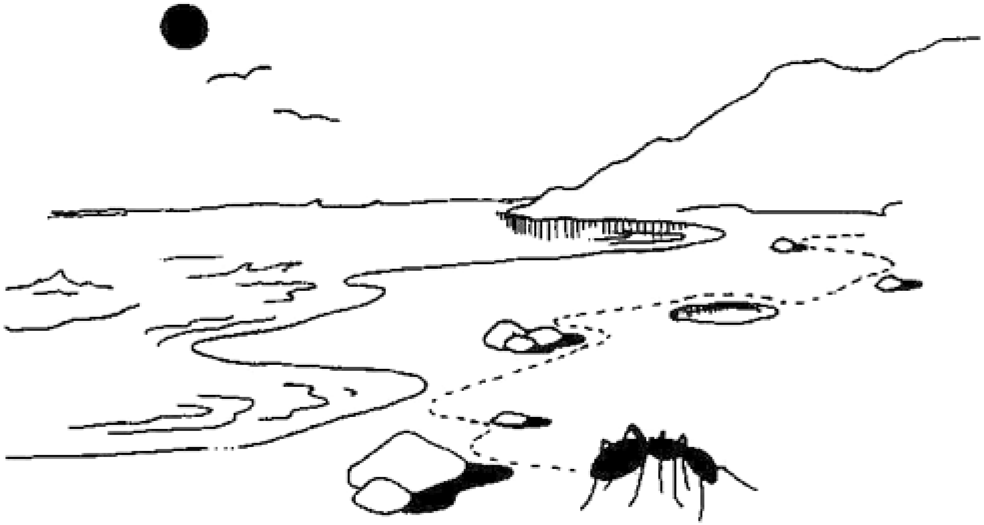
and to my wife

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT



[1]

An ant, viewed as a behavioral system, is quite simple. The apparent complexity of its behavior over time is largely a reflection of the complexity of environment in which it finds itself.

Herbert Simon

While watching ants wandering across the boulders on a beach, we may wonder what determines the complex behavior of those ants. Is it from the complexity of the environment or from the complexity of the ants? Nobel Laureate Herbert Simon argued that the cognitive and the physical capacity of ants are tiny compared with mighty nature and it was more from the complexity of the environment in his acclaimed book The Sciences of the Artificial [2]. He went on to explain how human problem solving was similar in the sense that the complexity of the problem largely determines the human problem solving process. He even wrote another paragraph exactly the same as the quote in the beginning except that “an ant” was replaced by “a man”.

What if the ants have a map of the beach? Knowing where the boulders are and where food may possibly lie may tremendously help the ants reach their goal. But how do they get such a map? Limited by its cognitive capacity, a single ant may be able to explore a small piece of the beach. While many ants explore some pieces of the beach, some may go faster and some may go slow. Some may get stuck by big boulders and some may get low hanging fruits quickly. The aggregate pattern of their exploration all together may provide hints of a beach, like where the obstacles are and where a flat terrain lies.

In the scope of this thesis, the ants are our students. The beach is the problem domain they are exploring, such as math and sciences. This thesis is our attempt to provide a method to find a more accurate map of the problem domain. Blessed with the increasing availability of student learning data and armed with the state-of-art machine learning tools, we can now stand five thousand miles above on the beach and have the telescope to study richly recorded ants' paths. The map starts to emerge and we then make it more accurate by learning from data.

I would like many people who have helped and supported my path to the completion of the research and my Ph.D. My college mentors Chunbai Zhang and Huiqiang Ge, and my graduate school mentors at UT Austin Susan Williams, Paul Resta, and Min Liu, prepared my early training for the endeavor. The Datashop team – Alida Stogsholm, Ben Billings, Kyle Cunningham, Brett Leber, and Sandi Demi – have provided me support on storing and acquiring data. Ajit Singh has provided essential help on the machine learning tool we used. Steve Ritter, John Steinhart, and Christy McGuire at Carnegie Learning have provided numerous support on Cognitive Tutor.

My committee member Brian Junker, Geoff Gordon, Noel Walkington, each from unique perspective, shaped my research into solid forms, teaching me how to draw strengths from different fields. My advisor Kenneth Koedinger has been mentoring me in the past five years. From him, I learned what is meant to be passionate for a field for life with an ultimate goal to benefit mankind, which is engraved in my heart.

I am in debt to my wife Ting Chen and my parents. With them, my happiness is doubled and sadness is halved.

ABSTRACT

In this thesis, we 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 brought in the concept of Q-matrix from Psychometrics 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 subject experts 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 cognitive model given extensive human labeling. We introduced the concept of P-matrix and developed a penalized combinatorial search built on top of the latent variable models. The search mechanism semi-automatically improves existing cognitive models by “smartly” choosing features from the P-matrix and incorporating them into the Q-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 used Exponential Principal Component Analysis that decomposes student-task matrix into a student-skill matrix and a skill-item matrix. We then compared its performance with LFA.

At the end of the thesis, we discuss several applications of LFA to improve student learning. 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 who are using the Tutor is more than two million hours per year in total.

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.

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 [3-5]. 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" [6-13]

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 is becoming available. Within the 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 set of production rules or skills encoded in intelligent tutors to model how students solve problems. (Production, skill, and rule are used interchangeably in this paper.) Productions embody the knowledge that students are trying to acquire, and allows the tutor to estimate each student's learning of each skill as the student works through the exercises. For example, the following table shows three skills used in the Area unit of a geometry tutor.

Table 1 Examples of the skills in a cognitive model

Skill name	Skill meaning
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.

The properties of the skills in a cognitive model contribute to a student's performance on solving items that require those skills. Figure 1 shows a visual representation of several items. On the right hand side are the items. Each item has responses as 1 for correct and 0 for incorrect.

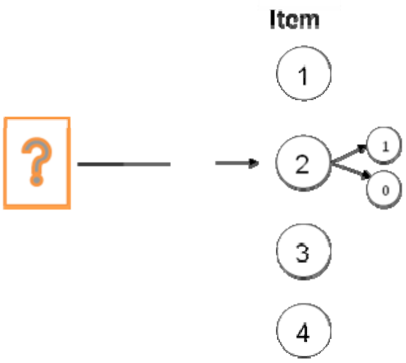


Figure 1 A graphical representation of student responses on items. The question mark represents the latent skills that determine student performance.

Discovering the set of skills can be formulated as a structure learning problem of uncovering the hidden layer of a multi-layer probabilistic graphic model with all variables being discrete. Unlike many standard machine learning problems with the goal of accurate prediction at the end, a unique requirement of this problem is that the skills discovered in the cognitive model need to be interpretable to human beings. 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. Tutor authors need to understand the skill labels to be able to author targeted items and hint messages. Being able to communicate the meaning of the discovered skills to humans is a crucial step for it to be useful. The unique side of the Learning Factors Analysis framework is that it 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 answers 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 skills?

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

Question 4 -- how to discover the latent skills or at least some properties of the skills 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 a 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 is then presented and three model operators on incorporating the information of the P matrix into the Q matrix. This chapter attempts to answer questions 2 and 3. In both Chapter 3 and Chapter 4, we apply LFA to real world data sets and show how it works.

Chapter 5 – We show how LFA can be used to answer different research questions. One example is using the AFM model to reduce over practice by students.

Chapter 6 -- We present a method call Exponential Family Principal Component Analysis to automatically extract Q matrices and compare their properties with LFA. This chapter attempts to answer question 4. We also compare the strengths and weakness of LFA and EPCA.

Chapter 7 – 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 [6-13] and latent factor models [14]. In data mining, it borrows the idea of improving the Q-matrix from [15, 16]. In statistics, particularly a branch of statistics called psychometrics, it shares strength with Q-matrix and item response models [17]. In cognitive psychology, it extends the early work in learning curve analysis [18-20]. LFA seamlessly puts the ideas from different fields into one framework and in some of those fields LFA makes a unique contribution and extension. The following sections show each relevant field 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 performance and the amount of practice [21] . Depicted by Eq. (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”, seen in Figure 2

$$Y = aX^{b} \qquad (1)$$

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

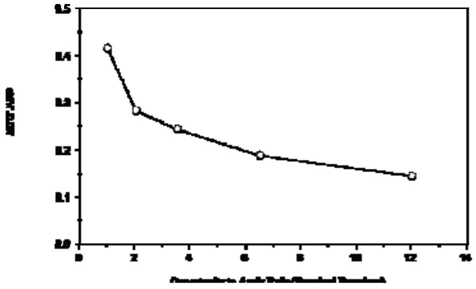


Figure 2 A power law learning curve

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 [21], as shown in Figure 3. They also found the power relationship holds if the complex skill can be decomposed into subskills, each of which exhibits a smoother learning curve.

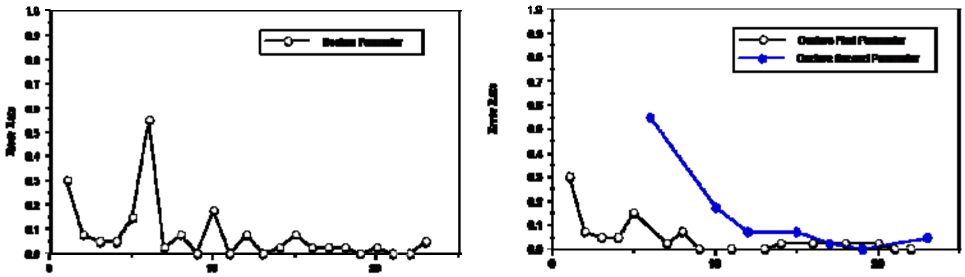


Figure 3 A learning curve with blips (*left*) split into two smoother learning curves (*right*)

As seen on the left in Figure 3, 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 provided a better fit to the data as shown on the right in Figure 3. 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 [22].

Item response models [17] 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 [23].

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 [19, 24-26] using theory refinement to introduce errors to model incorrect student behaviors [27], and using Q-matrix to discover knowledge structure from student response data [28].

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 indicate how skills contribute to student performance on the items.

Two extreme solutions to this problem are 1) to model the student responses with all the items (Figure 4), and 2) to model the student responses with a single latent factor, such as student intelligence (Figure 5). These two approaches represent the way that modern standardized tests are built.

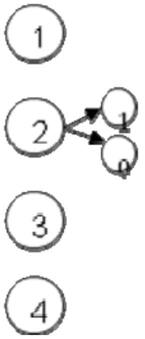


Figure 4 Item response model

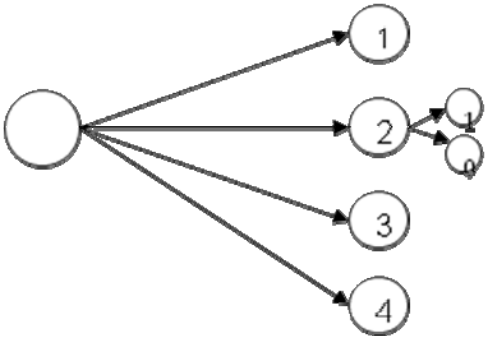


Figure 5 Single latent variable response model

One distinction between LFA and the previous two extremes is that LFA characterizes student responses on items in terms of the skills students use, i.e. the cognitive model, seen in Figure 6.

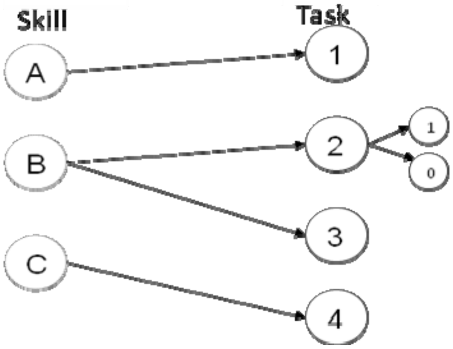


Figure 6 Cognitive model

Compared with the simulated student approach, our method does not require building a full-blown simulated student, although it can be argued our method is a very simple "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 found by adding/merging/splitting difficulty factors in our method. Table 2 sketches the differences between different methods. In this thesis, we also compare the performance of LFA with one machine learning approach called Exponential Principal Component Analysis [29].

Table 2 Comparing LFA with other 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	Can skip human encoding
DiBello et al.'s models	Yes	Yes				
Q-Matrix	Yes	Yes	Yes			
Draney, et al.'s model	Yes			Yes		
Corbett's			All manual	Yes	Yes	No
EPCA	Not IRT based. But account for students	Additive plus nonlinearity	Yes	Not yet	Depends	Can skip up front encoding
LFA	Yes	Yes	Yes	Yes	Yes	No

3. Learning Factors Analysis – The Static Part

3.1 The Q-Matrix

The Q-matrix is a Boolean matrix describing the relationship between items and skills [17, 30] . 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 3 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.

Table 3 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)

The power law model applies to individual skills and does not typically include student effects. Because typical cognitive model have multiple skills, and the data contains multiple students, following Draney, Wilson and Pirolli [31] we made four assumptions about student learning to extend the power law model.

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

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

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

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

Based on the assumptions, we developed a multiple logistic regression model to model the item responses given the skills, depicted by Eq. (2). It captures that the probability for student i to get item j right is proportional to how knowledgeable the student is θ_i plus for each skill needed for this item q_{jk} the "easiness" of that skill, plus an increment γ_k based on how much practice the student has had on that skill T_{ik} ,

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

where

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

θ_i = coefficient for student i

$\boldsymbol{\beta}$ = skill easiness coefficient vector

$\boldsymbol{\gamma}$ = skill learning rate coefficient vector

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 k s in item j in the exponent on logit[1] scale. 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, the logit of which is determined by a linear combination of student proficiency, skill easiness, and learning.

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 [17] with skills as the item attributes.

Ongoing work by Pavlik and Cen splits T_{ik} into the practice opportunities where the student get it right and the practice opportunities where the student get it wrong, which leads to the Performance Factor Model [32].

3.3 The conjunctive factor model (CFM)

One potential problem with AFM is the way it handles conjunctive skills. Suppose there is an item requiring two skills, as shown in Table 4. We would expect the item requiring two skills would be more difficult that the items with one of those skills. Suppose a student has a θ = 0; two skills above have β = logit(.8) and logit(.5) (which is 0); and there is no learning (γ = 0). In a conjunctive sense, we need a prediction of .4 (= .8 * .5). AFM will predict the third item with probability of .8 (=1/(1 + exp(-(logit(.8) + logit(.5)))), predicting a harder item is easier. One way to fix this problem is to have constraints on the parameter values on the AFM model so that the logit are shifted into a more multiplicative looking when the logit values are negative.

Table 4 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 Eq. (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 the same meaning as those in AFM. CFM and AFM reduces to the same form when there is only one skill per item.

[1] $\text{logit}(p) = \log \frac{p}{1-p}$

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

The conjunctive IRT model in Eq. 2 builds upon Embretson's multicomponent latent trait model (MLTM) [25], Dibello's Unified Model (UM)[19], and Davier's General Diagnostic Model (GDM) [18]. This model is also close to the frequentist version of the Noisy-Or Component model [33].

3.4 Parameter estimation

3.4.1.1 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) has good asymptotic properties (asymptotically unbiased, asymptotically efficient, and asymptotically normal) for estimators, under certain regularity conditions. In practice these properties appear to be approximately true, given a moderately large sample size. Given that we are only trying to learn a small number of parameters (see the discussion in Section 6.7), we used MLE to jointly estimate the student, skill, and learning parameters.

$$[\theta, \beta, \gamma] = \underset{\theta,\beta,\gamma}{\arg\max}\, LogLikelihood[\theta, \beta, \gamma; \mathcal{X}] \quad (4)$$

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

The Additive Factor Model likelihood function can be shown to be

$$LogLikelihood[\theta, \beta, \gamma; \mathcal{X}] = \sum_{i=1}^{n} (y_i z_i - \log(1 + e^{z_i})) \quad (5)$$

$$z_i = \mathrm{\beta}^T \mathbf{x}_i$$

The Conjunctive Factor Model log likelihood function can be shown to be

$$LogLikelihood[\theta, \beta, \gamma; \mathcal{X}] = \sum_{r=1}^{n} (y_r \log(p_r) + (1 - y_r) \log(1 - p_r)) \quad (6)$$

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

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

By doing an unconstrained optimization on the log likelihood function, we can get the student, skill, learning parameters from both AFM and CFM. However, as the number of parameters in AFM and CFM (mainly the number of student parameters) increase as there are new observations from new students, it is likely we are not in the asymptotic regime. This observation leads to the following method.

3.4.1.2 Penalized Maximum Likelihood Estimation

In earlier work, we found that freely maximizing the likelihood based on Eq. (5) and (6) often yielded student parameters that appear unreasonable. We hypothesized that it was caused by over fitting. We will talk about this issue in details at Section 6.7. To fight overfitting, we designed a Penalized Maximum Likelihood Estimation method (PMLE) [20], which penalizes the oversized student parameters in joint maximum likelihood estimation. Thus, PMLE maximizes the penalized likelihood depicted in Eq. (7). Maximizing the penalized likelihood in this equation 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.

$$ll_{PMLE} = ll_{MLE} - \frac{1}{2}\lambda\sum_{i=1}^{I}\theta_i^2,\ \lambda\text{=1 by default} \qquad (7)$$

where

I is the total number of students

In MLE, it is likely we are not in the asymptotic region. With PMLE, we push asymptotic region closer to us, in the hope of getting better out-of-sample results.

3.5 Assessment of the Statistical Models

Good statistical models balance between model fit & complexity minimizing prediction risk. They capture sufficient variation in data but are not overly complicated [34].

We choose two measures for model assessment -- K-Fold Cross Validation, shown in Eq. (8), which is time-consuming and more accurate estimate of prediction errors, and BIC, shown in Eq. (9) , which can be fast to compute but may be a fairly crude approximate of the prediction errors. It is worth noting some properties of BIC here. BIC is asymptotically consistent as a model selection tool, meaning that as sample size grows to infinity, BIC will choose the true model given the model space where the true model is included. If the data size is limited, often the case in educational data sets and social science data sets, BIC may prefer overly simple models [35]. However, the limit assumes that the number of parameters in each model is fixed, while the amount of data increases. Thus, these BIC theorems do not apply to AFM because the number of parameters (usually the number of student parameters) in AFM increases as the sample size increases. We use BIC mainly in the search process because the data set is held constant and it is much faster to compute BIC than to compute cross validation errors.

$$CV = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{f}^{-\kappa(i)}(x_i))^2 \qquad (8)$$

$$BIC = -2LogLikelihood + numParemeter * numObservation \qquad (9)$$

where $\hat{f}^{-\kappa(i)}(x_i)$ is the fitted function on the data with the kth fold removed.

3.6 Assessment of the Cognitive models

With AFM or CFM, we can then proceed to compare various cognitive models. The cognitive models with lower cross validation errors are the candidates of being better cognitive models. If the computation becomes issue, for example, in the search process described in the later chapter, BIC may be a substitute for the purpose of fast computation. The reason we call them candidates is that other criteria may also be considered such as the interpretability of the skills.

3.7 Example of AFM -- Geometry Area

3.7.1 Applying AFM

The data obtained from the Area Unit of the Geometry Cognitive Tutor (see http://www.carnegielearning.com and https://pslcdatashop.web.cmu.edu/). The initial cognitive model implemented in the Tutor had 15 skills that correspond to productions or, in some cases, groups of productions. Descriptions of these skills are in Table 5.

Table 5 A list of skills used in the initial cognitive model of the Geometry Tutor

Skill Name	Skill Meaning
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. This sample is a subset of the full data set from Datashop https://pslcdatashop.web.cmu.edu. 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 contains a unique anonymous identifier for each student. Success is whether the student did that step correctly or not in the first attempt. 1 means success and 0, failure. Step is the particular step in a tutor problem the students attempted. “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 6. 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

We fit AFM on the data and get the coefficients. The coefficient estimates for the skills and students, and the overall model statistics are summarized in the table below.

Table 7. 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 Opportunities is the average amount of practice per skill across all students. Initial Probability is the estimated probability of getting a problem correct in the first opportunity to use a skill across 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.

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

Student	Intercept
student0	1.18
student1	0.82
student2	0.21

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

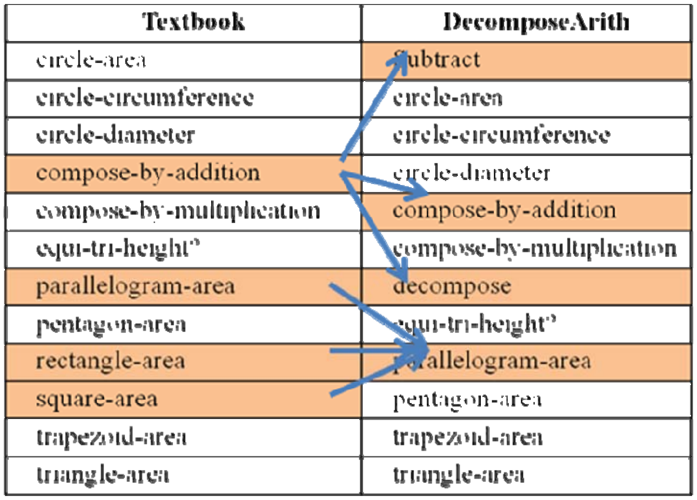
The higher the intercept of the each skill, the lower the initial difficulty the skill has. The higher the slope of the each skill, the faster students learned the skill. Pentagon-area is the hardest skill with the intercept of -2.16. Parallelogram-area is the easiest skill with the intercept of 2.14. Three skills have small slopes close to zero -- Compose-by-addition (-.04) and Parallelogram-area (-.01), Triangle-area (.03). Parallelogram-area was already mastered with an initial success probability .95. It appears that more practice on those skills does not lead to much learning gain. Interestingly, although PENTAGON-AREA is the hardest skill among all, it has the highest learning rate .45, leading to bigger improvement with more practice.

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

3.7.2 Comparing two cognitive models

Researchers hypothesized various cognitive models for this data set (see Datashop website https://pslcdatashop.web.cmu.edu/). Shown in Table 8, model Textbook is a simplified version of the original cognitive model. Model DecomposeArith splits "compose-by-addition" into "subtract", "compose-by-addition" and "decompose". Model DecomposeArith also combines "parallelogram-area", "rectangle-area" and "square-area" into "parallelogram-area". Notice "compose-by-addition" in model DecomposeArith is no longer the same as "compose-by-addition" in model Textbook as the former has less items associated with it. "Parallelogram-area" in model DecomposeArith is no longer the same as "parallelogram-area" in model Textbook as the former has more items associated with. Table 9 lists the statistics on the two cognitive models. By having a lower BIC, model DecomposeArith arguably describes the data better than model Textbook.

Table 8 Two cognitive models under comparison. The skills changed are highlighted. The arrows show the directions of change of the skills.



Textbook	DecomposeArith
circle-area	subtract
circle-circumference	circle-area
circle-diameter	circle-circumference
compose-by-addition	circle-diameter
compose-by-multiplication	compose-by-addition
equi-tri-height?	compose-by-multiplication
parallelogram-area	decompose
pentagon-area	equi-tri-height?
rectangle-area	parallelogram-area
square-area	pentagon-area
trapezoid-area	trapezoid-area
triangle-area	triangle-area

Table 9 Statistics of two cognitive models

	AIC	BIC	Log Likelihood	Number of Parameters
Textbook	5,167	5,710	-2,501	83
DecomposeArith	5,086	5,629	-2,460	83

3.8 Comparing AFM and CFM

To compare CFM with AFM, we used both a simulated data set and a real assessment data set. Since students in the assessment data set were not exposed to repetitive learning opportunities, we removed the learning term from both models. Cross validation errors and the interpretability of the actual parameter fits are used to evaluate the models. The details of the comparison can be found in [36].

The simulated data is used to answer the question "If the data is conjunctive, which model is better?" We simulated data drawn from a CFM model with 100 student parameters, 3 skill parameters, and 7 items. We explored four different sets of the three skill probability values (.1, .5, .9), (.1, .1, .1), (.4, .5, .6) and (.9, .9, .9). In nearly all cases, CFM-PMLE was as good as or better than AFM-PMLE in cross validation. The biggest difference was from the skill set (.9, .9, .9) because the skill parameter values are so high that AFM-P cannot behave in a conjunctive form (which it can if the logit values of parameter estimates β are negative). Table 10 shows the results from one of the above skill sets. As stated as the beginning of the paragraph, this is only a sanity check on whether CFM is able to fit the data if the data is conjunctive. We should not be surprised that CFM fits better. What is surprising is how well AFM does. AFM can model conjunctive data by having negative β parameter estimates, as is illustrated in the last column of Table 10. Thus there are two ways to show whether a real data set has conjunctive character. One is a better fit by CFM-P than AFM-P. The other is all β being negative in AFM-P.

Table 10 Model comparison of the simulated data. β = (.1, .5, .9).in probability. AFM-P stands for fitting the AFM with penalized MLE. CFM-P stands for fitting the CFM with penalized MLE.

	CV	CVSd	$\hat{\beta}$ in probability	$\hat{\beta}$ in logit
AFM-P	0.120	0.281	(0.03, 0.34, 0.73)	(-3.4, -0.67, 0.97)
CFM-P	0.111	0.174	(0.07, 0.5, 0.89)	(-2.54, 0.02, 2.07)

We explored these different possibilities in a data set used in a prior cognitive modeling work that predicts a (mostly) conjunctive structure [13]. The real data set EAPS is taken from a difficulty factor study of 247 U.S. algebra students. There are 1976 observations and 96 distinctive items. A simplification of their skill coding involves 3 skills is shown in Table 11. A sample of the Q-matrix is shown in Table 12. Notice certain items, such as "waiter-story-result-easy-mult", have no skills labeled in this Q-matrix due to the simplification.

Table 11 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 12 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
… …			

In the real data set, CFM-P is better than AFM-P by having a lower cross validation error, shown in Table 13. Notice AFM-P is essentially performing as a conjunctive model with all β estimates in logit being negative. This verifies the fact that the cognitive model underlying this data set has a conjunctive character.

Table 13 Model comparison of the EAPS data.

	CV	CVSd	$\hat{\beta}$ in probability	$\hat{\beta}$ in logit
AFM-P	0.202	0.142	(0.35, 0.47, 0.43)	(-0.63, -0.14, -0.3)
CFM-P	0.187	0.221	(0.61, 0.7, 0.67)	(0.43, 0.85, 0.7)

4. Learning Factors Analysis – The Dynamic Part

4.1 The P-Matrix

Section 4.1, 4.2, 4.3 are the innovative parts of Learning Factors Analysis to create a better cognitive model. First, corresponding to the Q-Matrix, we propose a new concept called P-Matrix. A Q-matrix is, in fact, a set of features of the items labeled by domain experts before it is put to use by students. A P-matrix is a set of features of items labeled by experts after a Q-matrix is put to use. 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 14, the second and the third item are labeled with the same set of skills. However, the third item may be 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 Table 15. In this example, we can put "Dealing with negative numbers" as one difficulty factor. The first two items have zero as the factor value and the third item has 1 as the factor value. If there is a fourth item "2*8+30", the expert may add a second factor "Two digit arithmetic" with 1s for the last three items in the P-matrix.

Table 14 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
2*8 +30	1	0	1	0

Table 15 A P-matrix

Item | Skill	Dealing with negative numbers	Two digit arithmetic	…
2*8	0	0	
2*8 - 3	0	1	
2*8 - 30	1	1	
2*8 +30	0	1	

4.2 Model operators

The second step to create a better cognitive model is to explore incorporating the information in a P-matrix into the existing Q matrix. We defined three model operators – "add", "merge", and "split" – to perform this function.

"Add" simply moves a column from P to Q. Table 16 is an example of adding the "Dealing with negative numbers" column in P to Q.

Table 16 Adding column "neg" in P to Q

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

"Merge" takes the Boolean operation Or among existing columns in Q. We denote the columns vectors to be merged as $\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_n$. The merged skill column vector $\mathbf{q}_{merged} = \mathbf{q}_1 \mid \mathbf{q}_2 \mid \mathbf{q}_n$, where | is the bitwise OR operator. Table 19 shows an example of merging the "add" and "sub" columns in Q.

Table 17 Merging "Add" and "Sub" in Q

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

"Split" refines an existing skill into two skills based on the presence of a factor. Splitting column vector $\mathbf{q}$ by column vector $\mathbf{p}$ creates a new column vector with values $\mathbf{q} \& \mathbf{p}$, where & is the bitwise AND operator, and turns the existing $\mathbf{q}$ into $\mathbf{q} \wedge \mathbf{p}$, where ^ is the bitwise XOR operator. Table 18 shows an example of splitting "Sub" in Q by "Neg" in P. "Sub&neg" is the result of "Sub" AND "Neg" and "Sub^Neg" is the result of "Sub" XOR "Neg". Notice "Split" and "Merge" do not change the conjunctivity of the Q-matrix while "Add" changes the conjunctivity of the Q-matrix.

Table 18 Splitting "Sub" in Q by "neg" in P

Item | Skill	Add	Sub^Neg	Mul	Div	Sub &neg
2*8	0	0	1	0	0
2*8 – 3	0	1	1	0	0
2*8 - 30	0	0	1	0	1
2*8 +30	1	0	1	0	0

A concrete example of "Split" is from the cognitive model with 15 skills described in Section 3.7. We tested a "split" on "triangle-side" by the factor of whether the side is a base or a height. Thus the original skill was split into "triangle-side-base" and "triangle-side-height". Now the new model has 16 skills. Shown in Table 19, the new model is not better than the original cognitive model in terms of BIC, suggesting the factor is not necessary.

Table 19 Model statistics after a split

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

4.3 Model search

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 (i.e. a Q-matrix), and a P-matrix, LFA 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 [37]. It starts from an initial node, iteratively creates new adjoining nodes, and explores them to reach a goal node. Factors in the P matrix are incorporated into the Q matrix through model operators. To limit the search space, it employs a heuristic to rank each node and visits the nodes in the order of this heuristic estimate. Either cross validation scores or BIC can be used as the heuristics in the search. We use BIC as an illustrating example in this document, because it is faster to compute and works as reasonably well shown in the next section. As shown in Figure 7, 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 new model 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.

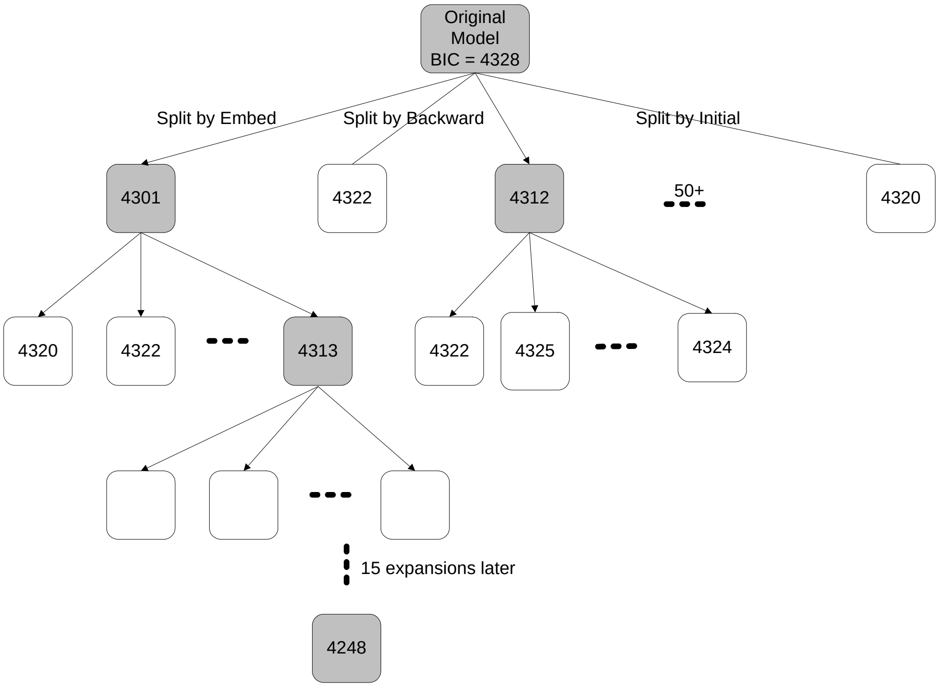


Figure 7 A best-first search through the cognitive model space

4.4 Example of the Search – Simulated Data

To test the effectiveness of the LFA search process and connect it with the comparison with EPCA (in section 6), we use a simulated data set, which is generated with AFM without the learning term, shown in Eq. (10). This AFM has 100 students, 3 skills and 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 β values as -2.2, 0 and 2.2, which correspond to probabilities of .1, .5, and .9. The True Q matrix is shown in Table 20. Each item has one skill involved. The P matrix contains the true Q matrix in it as well as another 21 fake factors. The goal is to see if LFA is able to recover the truth or some elements of the truth, if it starts searching from an empty Q matrix and the P matrix.

$$p_{ij} = \Pr(Y_{ij} = 1 \mid \theta_i, \boldsymbol{\beta}) = \frac{\exp(\theta_i + \sum_{k=1}^{K} q_{jk}\beta_k)}{1 + \exp(\theta_i + \sum_{k=1}^{K} q_{jk}\beta_k)} \qquad (10)$$

Table 20 The true Q matrix used to generated the data

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

Table 21 The P matrix

	A	B	C	AB	AC	BC	A1	A2	A3	A11	A22	A33	B1	B2	B3	B11	B22	B33	C1	C2	C3	C11	C22	C33
T1_100	1			1	1		1			1	1													
T2_010		1		1		1							1			1		1						
T3_001			1		1	1													1			1	1	
T4_100	1			1	1			1		1		1												
T5_010		1		1		1								1		1	1							
T6_001			1		1	1														1		1		1
T7_100	1			1	1				1		1	1												
T8_010		1		1		1									1		1	1						
T9_001			1		1	1															1		1	1

One parameter to determine in the model fitting and the search process is the penalization parameter λ. Table 22 and Figure 8 show the training errors and cross validation errors for the true Q matrix. When λ equals 1, the cross validation errors reach its minimum.

Table 22 Training errors and cross validation errors for the true Q matrix

Lamda	Training	CV
0	0.116	0.154
0.2	0.117	0.152
0.5	0.120	0.150
0.8	0.123	0.149
1	0.125	0.147
1.2	0.126	0.148
1.5	0.129	0.149

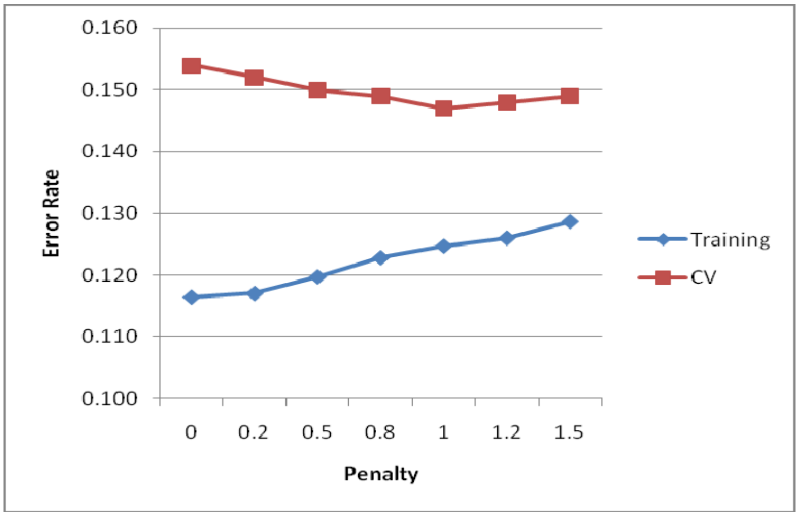


Figure 8 Training errors and cross validation errors for the true Q matrix

Table 23 shows the skill sets contained in the top three models found by LFA varied by the penalty parameter λ in the parameter fitting procedure. Table 24 lists the BICs for the true Q matrix (see Section 3.5 for the discussion of BIC) and for the top three Q matrices found by LFA. As λ is close to 1, the best Q matrix found by LFA (with skill A and C) is close to the true Q matrix (with skill A, B, and C). For the case that λ equals 1, the best Q and the third best Q found by LFA has the same low CV errors as the true Q, shown in Table 25 and Figure 10. Notice none of the best models found skill "B". Recall that in the simulated world, β for skill B is 0. When the Q matrix contains only skills C and A, the items with skill B (T2, T5, and T8) are still well predicted. With the BIC criterion in LFA search, which favors a simpler model, the Q matrix without skill B is more likely to be selected than the Q matrix with skill B.

The above analysis suggests a way to check the results found by LFA – After LFA returns a list of top models, use CV and expert judgment to find appropriate ones by comparing their prediction ability and the interpretability of the skill labels.

Table 23 LFA Search result 1 – the skills sets contained in the best, 2nd best, the 3rd best models found by LFA. Lamda is the penalty parameter.

Lamda	Best	2nd Best	3rd Best
0	A,AC	A,AB	A,B
0.2	A,AC	A,C	A,AC,B11
`	C,A	A,AC	A,AC,B11
0.8	C,A	A,AC	C,A,B11
1	C,A	C,AC	C,A,B11
1.2	C,A	C,AC	C,A,B11
1.5	C,A	C,AC	C,A,B11

Table 24. BICs for the true Q matrix and for the top three Q matrices found by LFA.

Lamda	trueModel	best	2nd Best	3rd Best
0	1,390	1,376	1,377	1,378
0.2	1,424	1,411	1,411	1,423
0.5	1,454	1,442	1,442	1,454
0.8	1,474	1,462	1,462	1,474
1	1,484	1,472	1,473	1,484
1.2	1,493	1,480	1,481	1,492
1.5	1,503	1,491	1,491	1,502

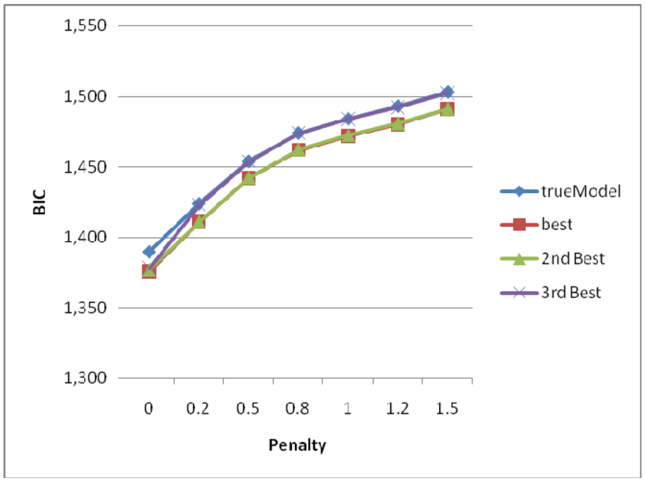


Figure 9 BICs for the true Q matrix and for the top three Q matrices found by LFA

Table 25 Cross validation scores for the true Q matrix and for the top Q matrices found by LFA when lamda equals 1

	Training	CV
TrueQ	0.125	0.147
Best	0.125	0.148
2nd Best	0.150	0.176
3rd Best	0.125	0.148

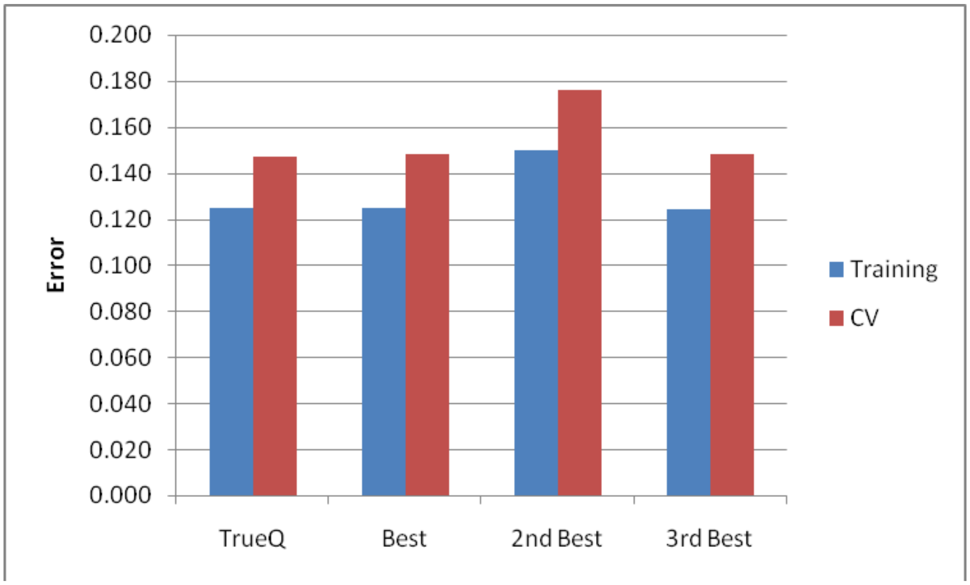


Figure 10 Training and cross validation error rates for the true Q matrix and for the top Q matrices found by LFA on the simulated data when lamda equals 1

4.5 Example of the Search – Geometry Learning Data

For the geometry data described in the previous section, we identified several factors for the P matrix. Here we list only the values of the factors in Table 26. The full list of factors is shown in Section 8.3. “Embed” indicates whether a shape is embedded in another shape. Consider two tutor problems requiring the same production rule CIRCLE-AREA at some step in the problem. In one of the problems, the circle is embedded in a square; while in the other one, the circle is presented alone. Students may find it harder to find the area of circle when it is embedded in another figure because extra effort is necessary to find the circle and its radius. “Backward” means whether the production rule to be used is in its backward form of a taught formula, or its forward form. The forward form of Compose-by-addition is S = S1 + S2, and its backward forma is S1 = S - S2. “Repeat” indicates whether the production rule has been used previously in the same problem. “FigurePart” indicates the part of the figure in the geometry shape to be computed.

Table 26. Factors for the geoemtry data

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

Table 27 lists the improved models found by LFA. The skills unchanged from the original Q are omitted for clarity. The improved skills common to most of the better models are Compose-by-multiplication, Compose-by-addition, Circle-area, and Triangle-area. All the new models suggest splitting Compose-by-multiplication into two skills – CMarea and CMsegment, making a distinction of the geometric quantity being multiplied.

Table 27. Top three improved models found by LFA with BIC as the heuristic.

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
BIC: 4,248.86	BIC: 4,248.86	BIC: 4,251.07

We also used LFA to answer the question are some skills 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-AR; 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 – CircleAR, CircleCD, Compose-by-addition, Compose-by-multiplication, Parallelogram, Pentagon, Trapezoid, and Triangle.

Then we substituted the original skill names with the new skill name in the data, ran LFA including the factors. The improved models by LFA with BIC are summarized below.

Table 28. 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
CircleAR *area CircleAR *radius*initial CircleAR *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. CircleAR is split into CircleAR *backward*initial, CircleAR *backward*repeat, CircleAR*forward, 2. Compose-by-addition is not split	All skills are the same as those in model 1 except that 1. CircleAR is split into CircleAR *backward*initial, CircleAR *backward*repeat, CircleAR *forward, 2. Compose-by-addition is split into Compose-by-addition and Compose-by-addition*segment
BIC: 4,169	BIC: 4,171	BIC: 4,171

LFA refines skill Circle, suggesting the distinctions made in the original model are necessary. In model 1, CircleAR is split into CircleAR*area, and CircleAR*radius. The other two models split it into CircleAR*backward, and CircleAR*forward, which are equivalent to CircleAR*area, and CircleAR*radius because of the one-to-one relationship between forward and area and between backward and radius. Thus, LFA fully recovers the two Circle skills and further refines one of them.

None of the models recovered skill Circle-CD, Trapezoid, Triangle or Parallelogram. This suggests that distinctions made in the original model are not necessary.

Skill Compose-by-addition is split into two skills by whether the composition was done on area or on segment, suggesting a refinement not anticipated in the original model.

5. Applications of LFA

As evidence of the utility and expressiveness of the LFA model and algorithm, we note that many researchers have used LFA as a new research tool to answer questions in domains beyond math and Cognitive Tutors.

5.1 Other Researchers' Use of LFA

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 [38].

Nwaigwe (Carnegie Mellon University) and colleagues at the University of Pittsburgh used the AFM component of LFA to explore the quality of different methods for analyzing student errors during training [39].

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 [40, 41].

One application we did with the AFM component of LFA was to use it to discovery over practice and under practice in the tutoring environment with the goal to improve student learning efficiency [42]. This work is described in the next section.

5.2 Improving Student Learning Efficiency by Reducing Over Practice

5.2.1 Discover Learning Inefficiency through AFM

By applying AFM 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) skills 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. "rectangle-area" is an example. This skill characterizes the skill of finding the area of a rectangle, given the base and height. As shown in Figure 11, students have an average initial error rate around 12%. After 18 times of practice, the average 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 skills include square-area and parallelogram-area. On the other hand, some difficult (i.e. low

β_j) skills 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 skill is needed for students to reach mastery. Other under-practiced skills include pentagon-area and triangle-area.

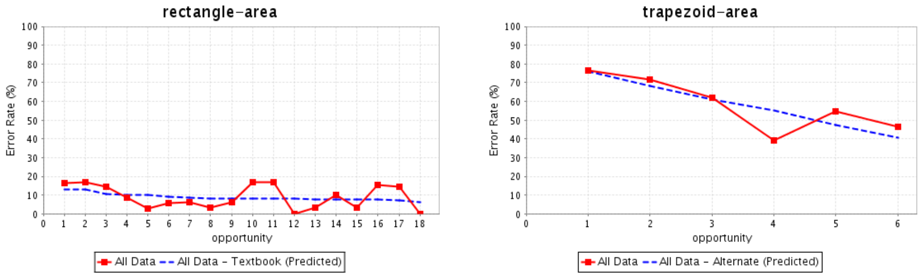


Figure 11 Learning Curve of Rectangle-Area and Trapezoid-Area – The solid lines are the actual error rates over the ordered number of practices. The dotted lines are the error rates predicted by LFA.

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 skills [13, 43]. Based on these estimates, the Tutor chooses to give students the problems with the skills students need to practice more. Table 29 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 29 column 3. We hypothesized that by using recalibrated the Knowledge Tracing parameters, the Tutor could saved significant learning for students. 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 29 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

We grouped the skills in the 2006 Tutor into several homogeneous groups according to their degrees of over-practice from the data collected in 2005. 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.

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

The under-practiced skill 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 skills (circle-circumference, trapezoid-area, trapezoid-perimeter, triangle-perimeter) are set to $P(L_0) = 0.5$.

The moderately-over-practiced skills (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 skill for information extraction are set to $P(L_0) = 0.9$.

5.2.2 Saving Student Learning Time while Maintaining Learning Gains

In a controlled experiment with 110 students in a high school near Pittsburgh, we found that students using the optimized tutor learned as much as the control group but in less time. As seen in Figure 12, the two groups have similar scores in both the pre test, the post test, and the retention test. There are 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.

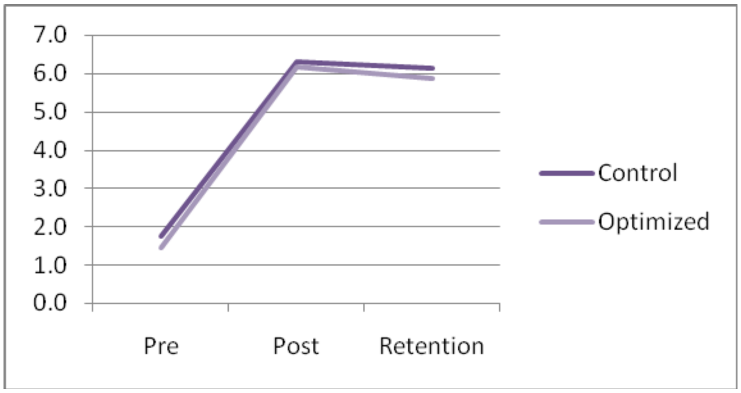


Figure 12 Pretest and post test scores over the two conditions (left) and the retention test scores (right)

The actual learning time in many units is significantly reduced for the students in the optimized group. As shown in Table 30, 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, a 12% reduction in the total tutoring time.

Table 30 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%		

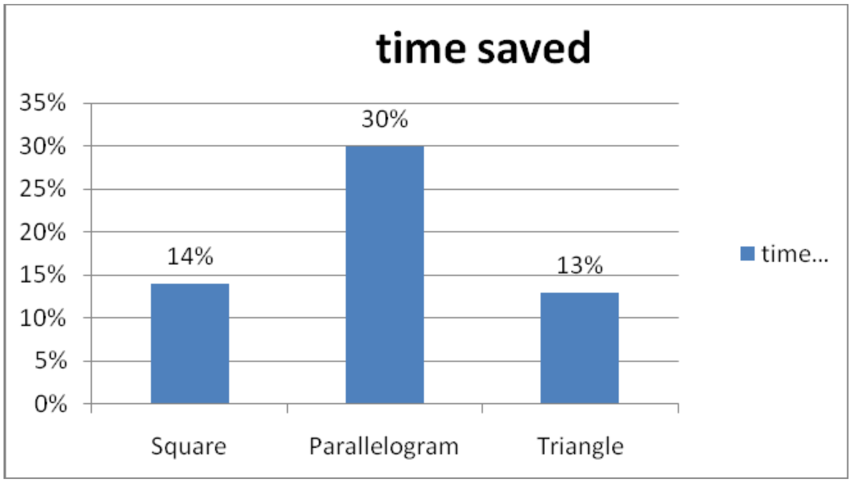


Figure 13 Percentage of Time Saved

6. Automatic Discovery of Q Matrices with EPCA

Like the secret key to the gate of treasure, the Q matrix serves as key to predict student performance on questions. LFA depends on the existence of such a matrix. The origination of such matrices involves extensive human expertise. Automatic discovery of Q matrices may significantly reduce the human labor in labeling P matrices. Exponential-Family Principal Component Analysis is one of the methods that attempt to solve this problem. We discuss EPCA in this section in detail. Another approach is Partial Ordered Knowledge Structures (POKS) [44].

6.1 Principal Component Analysis (PCA) and Exponential-Family Principal Component Analysis (EPCA)

Principal Component Analysis (PCA) is a popular method for feature extraction and dimension reduction. It is traditionally viewed as 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 [45].

EPCA, a generalization of PCA, addresses the problem when the noise is not Gaussian. The general idea of EPCA is that it views each data point $X_{ij} \in \mathrm{R}^d$ 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} [46].

$$\begin{aligned} & X_{ij} \sim P(\theta_{ij}) \\ & \theta_{ij} = U_{i.} * V_{.j} \\ & \Theta = UV \end{aligned} \qquad (11)$$

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 [29] and Singh [15] estimates Θ by alternatively optimizing the loss function while holding one of the U, V matrices constant one at a time.

6.2 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 aggregating the student performance data on students and questions, we get a student-item matrix, the cell values of which can be thought as generated from a binomial distribution. We propose four formulations of EPCA with increasing complexities to answer various research questions. For each formulation, we show the factored matrices as well as the corresponding optimization form. The advantages and disadvantages of each formulation are discussed.

$$\begin{aligned} X_{ij} &\sim Bernoulli(\theta_{ij}) \\ E[X_{ij}] &= \text{logit}[\theta_{ij}] \end{aligned} \tag{12}$$

6.2.1 Formulation 1

$$\begin{pmatrix} \beta_{11} & \beta_{12} & \beta_{13} \\ \beta_{21} & \beta_{22} & \beta_{23} \\ \beta_{31} & \beta_{32} & \beta_{33} \\ \beta_{41} & \beta_{42} & \beta_{43} \\ \text{M} & \text{M} & \text{M} \end{pmatrix} \begin{pmatrix} q_{11} & q_{21} & q_{31} & \text{L} \\ q_{12} & q_{22} & q_{32} & \text{L} \\ q_{13} & q_{23} & q_{31} & \text{L} \end{pmatrix}$$

$$\beta_{ij,} q_{ij} \in \text{R}$$

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

This is a direct translation of the model to the problem. The existing implementation of EPCA handles this formulation. However, due to the large number of elements of in U and V, the estimation errors could be large. The interpretation of the entries in the factored matrices is not obvious.

6.2.2 Formulation 2

$$\begin{pmatrix} \theta_{1} & \beta_{11} & \beta_{12} & \beta_{13} \\ \theta_{2} & \beta_{21} & \beta_{22} & \beta_{23} \\ \theta_{3} & \beta_{31} & \beta_{32} & \beta_{33} \\ \theta_{4} & \beta_{41} & \beta_{42} & \beta_{43} \\ & \text{M} & \text{M} & \text{M} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ q_{11} & q_{21} & q_{31} & \text{L} \\ q_{12} & q_{22} & q_{32} & \text{L} \\ q_{13} & q_{23} & q_{31} & \text{L} \end{pmatrix}$$

$$\theta_{i}, \beta_{ij,} q_{ij} \in \text{R}$$

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

This formulation adds a student parameter column to U and one 1s row to the skill-step matrix V accounts for student proficiency. However, it is still hard to interpret the meanings of entries other than the student proficiency.

6.2.3 Formulation 3

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

$$\theta_i \in \mathrm{R}$$
$$\beta_{ij} \le 0$$
$$q_{ij} \ge 0$$

$$\min Loss(U,V;X)$$
$$\beta_{ij} \le 0$$
$$q_{ij} \ge 0$$

Based on formulation 2, this formulation constrains β and q. Matrix V starts to behave more like a traditional Q matrix and β has a similar meaning to skill difficulty.

6.2.4 Formulation 4

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

$$\theta_i \in \mathrm{R}$$
$$\beta_{ij} \le 0$$
$$0 \le q_{ij} \le 1$$

$$\min Loss(U,V;X) + c\sum_{ij} \beta_{ij}{}^2$$
$$\beta_{ij} \le 0$$
$$0 \le q_{ij} \le 1$$

Based on formulation 3, this formulation constrains β to be negative and q to be in the range of 0 and 1 such that V is even closer to a Q matrix.

6.3 Complications of applying EPCA to real data

Oftentimes 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 lead to erroneous results. One solution to that is to add a weight matrix to the student-item matrix. If the student has done the item, the weight is 1. Otherwise the weight is 0.

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

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

6.4 Evaluation of EPCA – the Fold-in Algorithm

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 example illustrates the Fold-in algorithm that PCA uses to predict performance, given new items.

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

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

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

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

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

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. Using just a single point estimate of v_fold is an approximation, which may not be valid in our particular case (see the discussion in Section 6.7).

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

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

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

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

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

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

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

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

Step 4 Compare the actual student performance vs. the predicted student performance using

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

6.5 Connections between AFM and EPCA

Recall that AFM without learning takes the following form

$$\log\frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_{k=1}^{K} q_{jk}\beta_k \qquad (13)$$

This form can also be viewed from a matrix factorization point of view, shown in Eq. (14).

$$\begin{aligned} &\log P/(1-P) = UV \\ &U = [\theta \quad \mathbf{1}] \\ &V = [\mathbf{1}; \quad \beta Q] \end{aligned} \qquad (14)$$

where

P is an ngm matrix with p_{ij} as elements;

$/$ is an element-wise division operator;

U is the student-skill matrix with the student parameter θ as the first column and ones as the second column;

V is the skill-item matrix with ones as the first row and the vector matrix product of the skill difficulty vector β and the Q matrix as the second row.

Several observations can be made from Eq. (14). First, Eq. (14) is in the EPCA form and thus AFM without learning is a special case of EPCA. Second, both U and V are of rank 2. Third, we cannot tell the difference between two Qs whose row spans both contain the desired item difficulty vector βQ if $\beta_1 Q_1 = \beta_2 Q_2$.Then for any given student skill vector θ, we make exactly the same predictions with β_1, Q_1 as we do with β_2, Q_2. From these observations, if the data is truly generated from an AFM model without learning, EPCA will find at most two skills.

6.6 Results

6.6.1 Simulated Data

The data used here is the same data used in Section 4.4.

Given the existing implementation of EPCA, we used the first formulation described above, using a leave-one-out cross validation. There are two parameters we need to determine in EPCA. One is the regularization parameter and the other is the number of latent skills. In Figure 14, we have plotted the error rates from a combination of regularization parameters and the number of latent skills. In those plots, the presence of two or three skills leads to the lowest cross validation errors in EPCA. The lowest cross validation error 0.169 occurs when the regularization parameter equals 1 (although an even higher regularization parameter may be beneficial) and the number of skills equals 2. It makes sense because the V matrix from AFM has a rank 2.

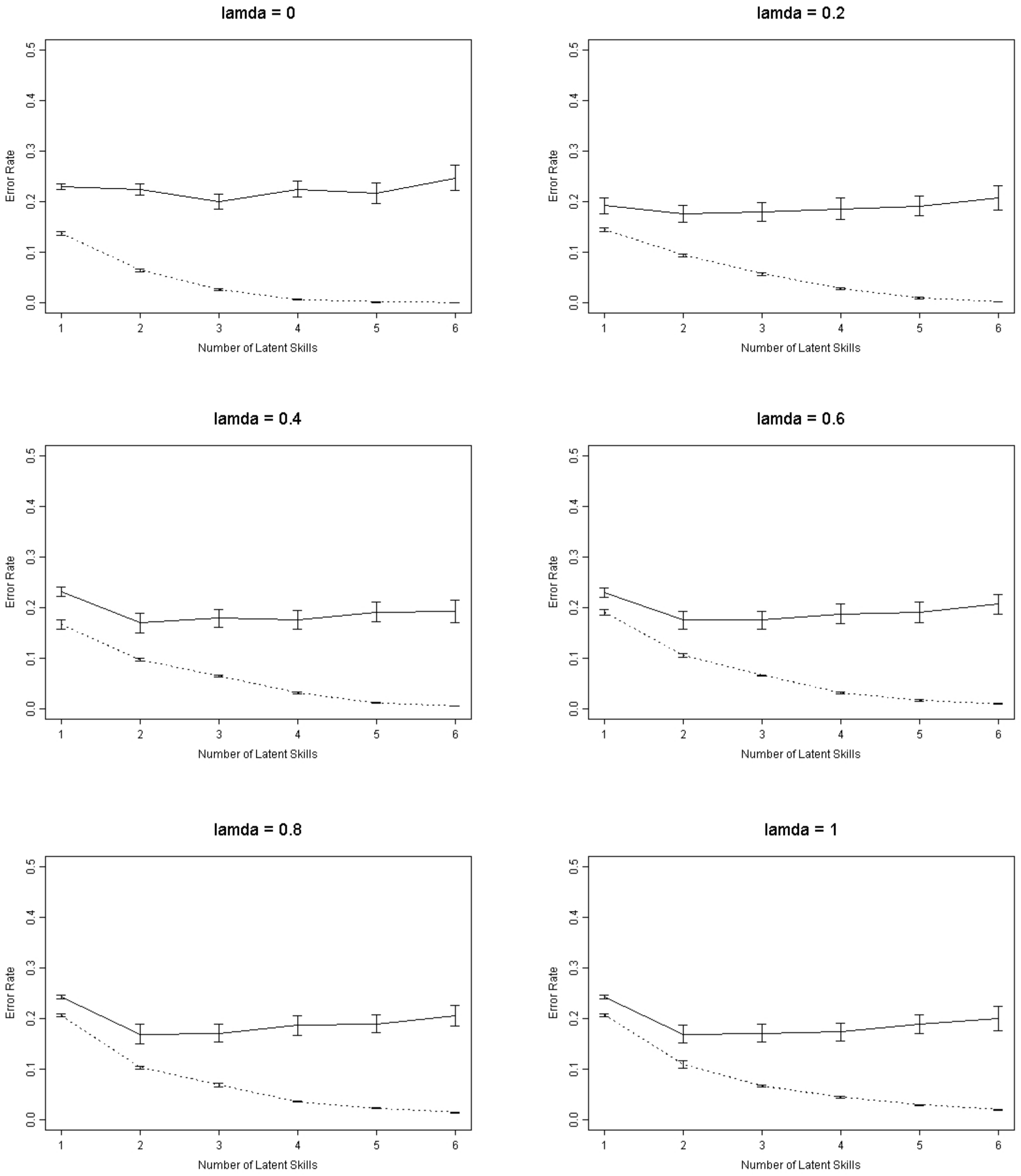


Figure 14 Cross validation errors, training errors and their standard errors. Each curve is plotted as a function of the number of latent skills. The solid line is for the mean cross validation errors and the dotted line is for the training error. One standard error bars are imposed on each error curve. The top left, top right, to middle left, middle right, bottom left and bottom left plots have the regularization parameters lamda for 0, .2, .4, .6. ,8. 1.

The following two tables show two Q matrices from two cross validation tests on the same date set when EPCA has the lowest cross validation error. In the Q matrix with cross validation on item 1, the item 1 portion of the matrix is not estimated because item 1 was omitted in the construction of V in EPCA. For a similar reason, the item 2 portion of the Q matrix with cross validation on item 2 is not estimated. The Q matrixes resulting from EPCA vary from one to another in each validation set from each test. It is worth noting that EPCA returns different results on the same cross validation set if we change the number of iterations or the starting point for the parameter estimation procedure, because the corresponding optimization problem for EPCA is not convex and has many local optima [29].

Table 31 Two Q matrices from EPCA on the same data

	Cross Validation on item 1		Cross Validation on item 2	
	A	B	A	B
T1_100			3.4	-0.4
T2_010	0.0	1.6		
T3_001	2.9	1.8	-1.7	-0.3
T4_100	-1.3	-2.7	3.1	-2.2
T5_010	-2.2	3.0	2.4	3.7
T6_001	0.7	3.7	-1.4	2.9
T7_100	-1.3	-1.3	2.1	0.2
T8_010	3.1	-0.3	2.8	-1.3
T9_001	2.1	2.8	-1.9	2.1

It is interesting to compare the best Q matrices found by EPCA and LFA. Shown in Figure 15, the cross validation error from EPCA is very close that from LFA, although EPCA does not require any human inputs.

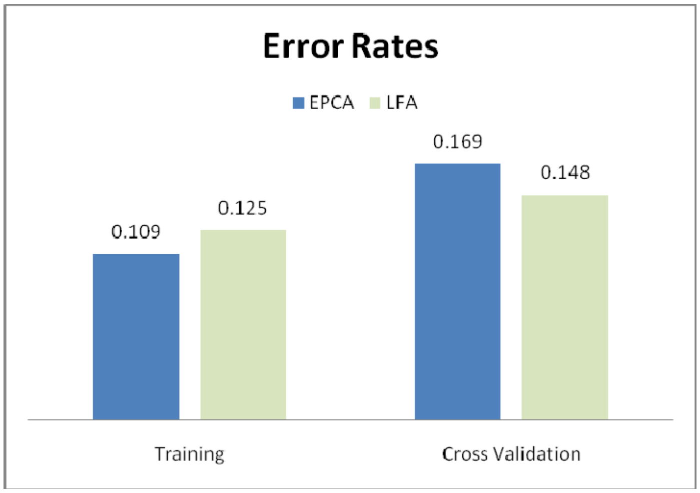


Figure 15 The error rates for the best Q matrices found by EPCA and LFA. In each group, the left bar is for EPCA and the right bar is for LFA.

The results here suggest that EPCA can be used as a heuristic to determine whether a data set is rank 2 and thus may conform to the unidimensionality assumption behind AFM (and the Rasch model more generally). As far as the interpretability of the skills discovered, EPCA may still require human subject experts to assign meanings to the skill labels.

6.6.2 Real Assessment Data

A fair comparison of EPCA and LFA would be on a real data set when the true Q matrix is unknown. Using a section of the data set described in Section 3.8 with 171 students and 96 items, we compared the performance of EPCA and LFA.

One thing worth noting is the data sparsity issue in this data set. Every student only did 8 out of the 96 items, causing most cell values in the student-item matrix to be empty. The researchers (Koedinger, personal communication) report that there was only class time for an 8 item quiz and they choose to sample more broadly within the item space (systematically generated from a hypothetical set of skills) rather than just choosing 8 items from this space for all student quizzes. Every item was seen by an average of 14 students with a minimum of 8 students and a maximum of 23 students. EPCA handles the data sparsity by using a weight matrix described in Section 6.3. However the sparsity imposes an issue when we evaluate the performance of EPCA with the fold-in algorithm. Recall in step 2 of the fold-in algorithm, "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". In order to test for up to 5 skills, we need to include 60% of the student observations, which gives us 5 data points for items with the minimum number of students (8) and 14 data points for items with the maximum number of students (23).

Shown in Figure 16, the lowest cross validation error 0.23 occurs for EPCA when the regularization parameter equals 1 and the number of skills equals 2 (although the regularization parameter > 1 could be beneficial). When the regularization parameter equals .2, .4, .6, and .8, the corresponding lowest CV errors occurred at 1, 1, 1, and 2 skills.

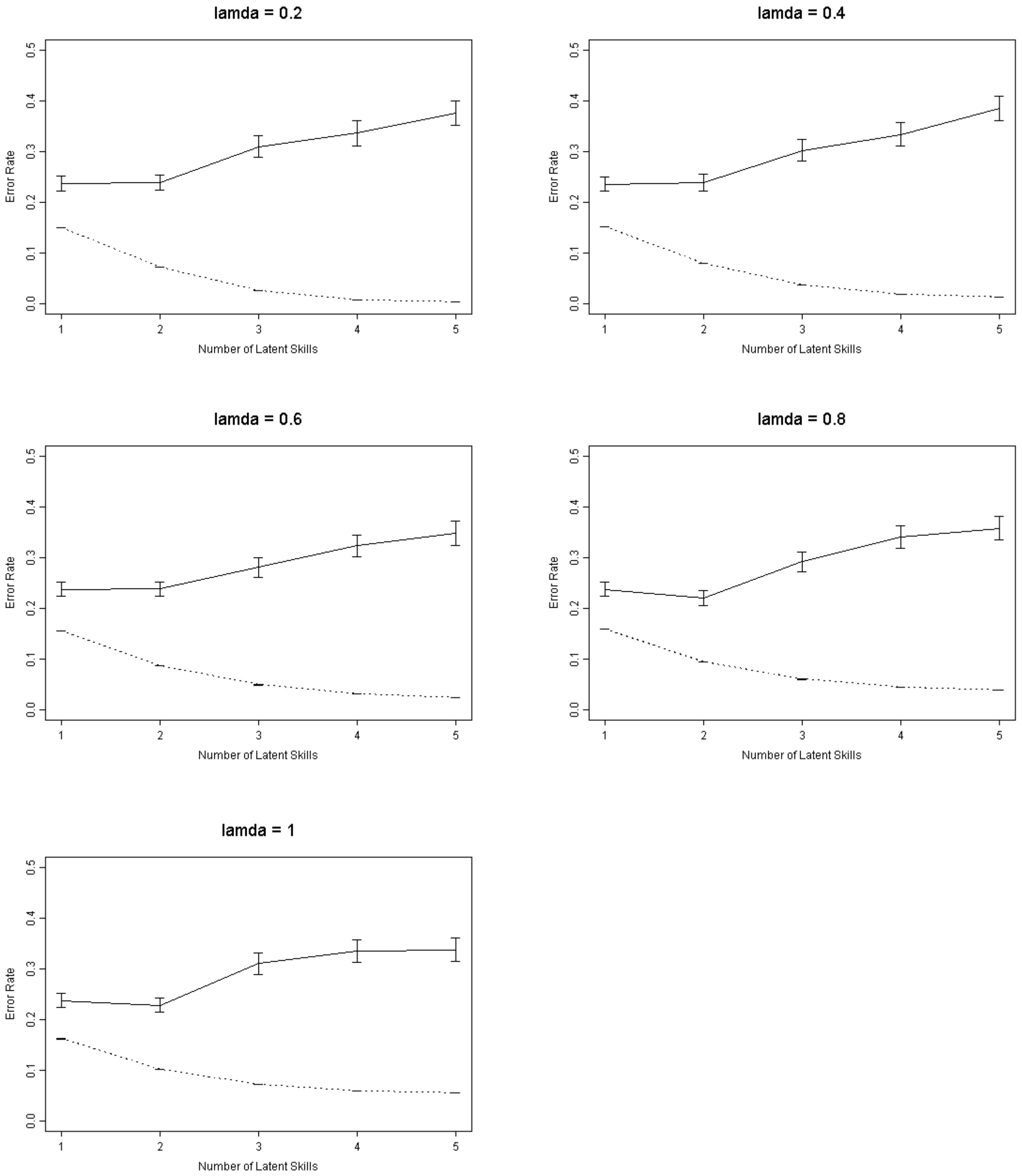


Figure 16 Cross validation errors, training errors and their standard errors by EPCA on the EAPS data. Each curve is plotted as a function of the number of latent skills. The solid line is for the mean cross validation errors and the dotted line is for the training error. One standard error bars are imposed on each error curve. Lamda is the regularization parameter.

Table 32 shows the lists of skills from the best three Q matrices found by LFA search with penalized AFM on the EAPS data from 23 factors listed in Section 8.4 with a regularization parameter of 1. BIC was used as the search heuristic. Lists of skills from the best three Q matrices found by LFA search with penalized AFM on this data are listed in Section 8.5.

Table 32 The lists of skills along with their parameter estimates from the best three Q matrices found by LFA search with penalized AFM on the EAPS data, ranked by BIC. The cross validation errors of the three models are listed in the last row.

Best Model	2nd Best	3rd Best
unknownPosition-result (1.15)	unknownPosition-result (1.11)	unknownPosition-result (1.18)
presentation-equation (-1.15)	presentation-equation (-0.99)	presentation-equation (-1.25)
numDifficulty-easy (0.85)	numDifficulty-easy (0.78)	numDifficulty-easy (0.89)
origArith-div (-0.66)	origArith-div (-0.71)	origArith-div (-0.62)
coverStory-lottery (0.62)	coverStory-lottery (0.43)	coverStory-lottery (0.53)
finalArith-div (0.35)	finalArith-div (0.3)	finalArith-div (0.39)
numCategory-hard-bball (-0.56)	numCategory-hard-bball (-0.68)	numCategory-hard-bball (-0.56)
	presentation-story (0.33)	presentation-word (-0.3)
0.172	0.172	0.171

Figure 17 shows the scatter plot of the values of the V matrix found by EPCA when the regularization parameter equals 1 and the number of skills equals 2. Each item on the plot is assigned a difficulty level between 1 (easiest) to 4 (hardest) according to their item features (verbal-result, verbal –start, equation-result, equation-start). There seems to be clusters of items according to their difficulty.

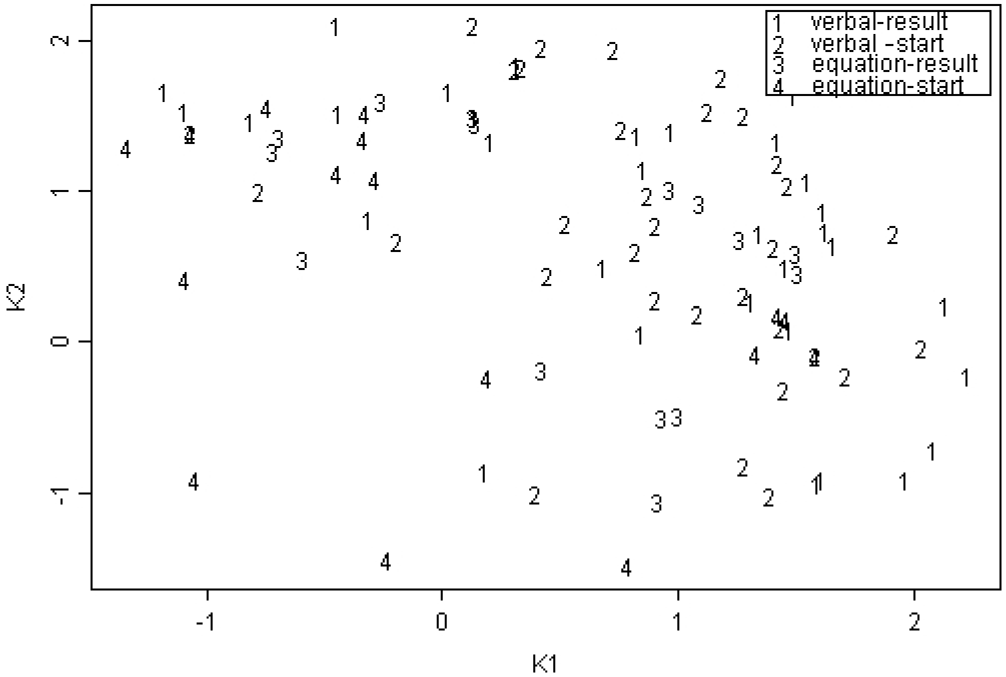


Figure 17 Scatter plot of the values of the V matrix found by EPCA when the regularization parameter equals 1 and the number of skills equals 2. The items are labeled with different numbers.

Table 33 shows a part of the Q matrix discovered by LFA on the EAPS data.

Table 33 A part of the best Q matrix found by LFA search on the EAPS data

Item	unknownPosition-result	presentation-equation	numDifficulty-easy	origArith-div	coverStory-lottery	final Arith-div	numCategory-hard-bball
bball-equation-result-easy-div	1	1	1	1	0	1	0
bball-equation-result-easy-mult	1	1	1	0	0	0	0
bball-equation-result-hard-div	1	1	0	1	0	1	1
bball-equation-result-hard-mult	1	1	0	0	0	0	1
bball-equation-start-easy-div	0	1	1	0	0	1	0
…							

Figure 18 shows the training errors and cross validation errors of EPCA and LFA on the best Q matrix. It is not bad in terms of prediction by EPCA, even if it does not require up front human labeling. The use of MAP (maximum a posteriori) estimates instead of Bayesian reasoning is the probable cause of the difference in train & test performance. The next section illustrates this point in detail.

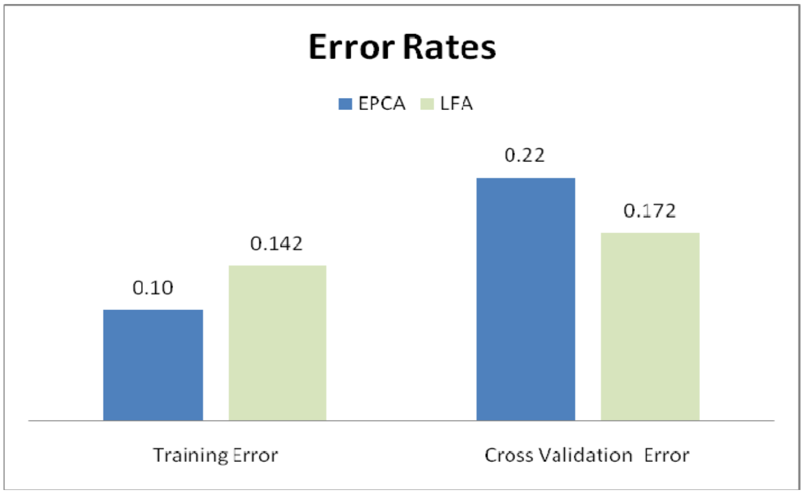


Figure 18 The error rates for the best Q matrices found by EPCA and LFA for the real assessment data set. In each group, the left bar is for EPCA and the right bar is for LFA.

6.7 Thoughts on MLE and Full Bayesian Modeling for EPCA

Although regularization is used along with EPCA, we observe that EPCA often leads to lower training errors but higher prediction errors. In practitioners' words, there is some amount of over-fitting going on, regardless of regularization. In this section, we explore the causes of over-fitting and discuss how to avoid over-fitting at all.

EPCA is computed via a maximum likelihood approach. Asymptotically, MLE estimators are consistent, efficient and optimal [47]. Suppose the data matrix X is dense with ngm entries, which are the number of data points. The factored U and V have ngk and kgm entries respectively, which are the number of parameters. In AFM without learning, there are only $n+k$ parameters. Compared with AFM, EPCA is less likely to compute the parameters in the asymptotic region, leading to inconsistent and inefficient MLE estimators. With regularization, it is possible to get EPCA to be closer to the asymptotic region, but not completely.

For a general model, having regularization on the model can be viewed as finding the MAP (maximum a posterior) estimate, the mode of the posterior distribution of the parameters. Denote θ as the parameter vector and X as the data. The maximum likelihood estimate is

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \Pr(X \mid \theta) \tag{15}$$

The MAP estimate is

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \Pr(X \mid \theta) \Pr(\theta) \tag{16}$$

When $\Pr(\theta)$ is a constant, the MAP estimate is the same the MLE estimate. However, the MAP estimate still does not capture the full uncertainly of the posterior distribution.

Figure 19 (courtesy of Geoff Gordon, taken from his graduate AI course slides) shows the MAP estimate (the red line) and a sample of the posterior distribution (the green lines) from the Irises data set. When the predictor's value is in the middle of the range, MAP is approximate the average of the posterior distribution samples. When the predictor's value is close to the extremes, MAP tend to make under prediction or over predictions.

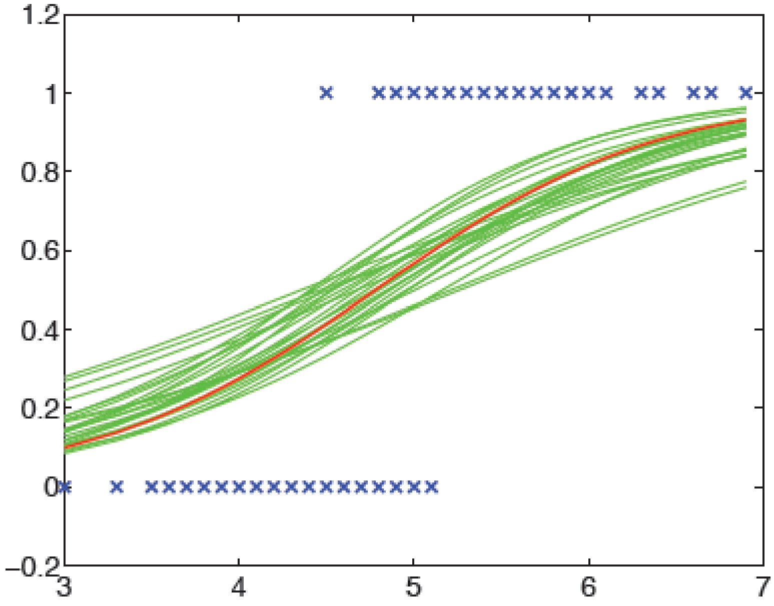


Figure 19 the MAP estimate (the red line) and a sample of the posterior distribution (the green lines) from the Irises data set

To fully caputre the uncertainty of the posterior distribution, we need a full Bayesian treatment. For EPCA, the inference needs to be done on U and V. Figure 20 is the plate view of the graphic model representation of EPCA with hierarchical priors. X is the ngm data matrix. U and V are the factored matrices with ngk and kgm elements. For EPCA to have priors, now, U depends on prior parameters μ, a vector of k components. Similarly, V depends on prior parameters λ, a vector of k components.

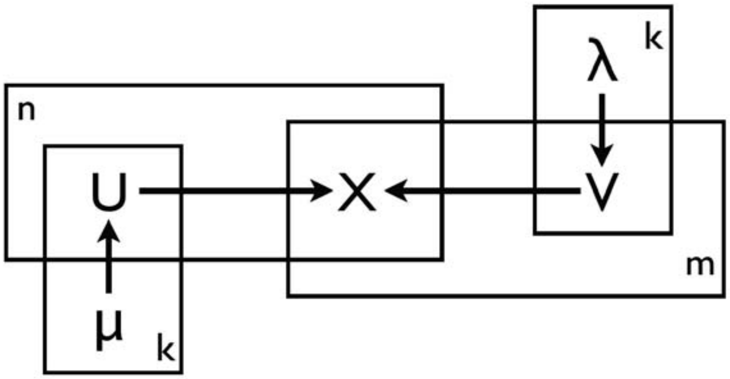


Figure 20 the plate view of the graphic model representation of EPCA with hierarchical priors

. To get the posterior distributions of U and V, given X, we can use an MCMC algorithm to get a sample of (U , V) pairs. When we need to make predictions, we sum over the samples to get the Bayesian prediction. The result of the full Bayesian treatment is that the training error and the test error are much closer together, and hopefully lower test error, although it is hard to know in advance how much lower it would be.

7. Conclusions and Future Work

While life might be simpler if we simply choose a method that generates the lowest cross validation errors, the two methods presented so far have different strengths and weaknesses. They cater to different purposes. The biggest strengths of LFA are that it brings human expertise into the discovery process. The labels it incorporates into the cognitive models are interpretable. Thus curriculum designers can use the meanings of the labels to author new items and write targeted hint messages. The price for the interpretability is that it requires human work to look at items and come up with features. EPCA is on the opposite side – it requires no initial human input. However, the factored matrices it produces have no label meanings and still require human expertise to assign meanings to the labels. This is not necessarily a minus because the factored matrices can be used entirely without labels. One application is to suggest good items for students to try next (similar to the case where collaborative filtering is used for online goods recommendation).

Keen readers may ask when we should use which tool. If we are blessed with an existing cognitive model and subject experts are willing to donate some time to improve the cognitive model, which is often true as well, it is good to have experts come with features and run LFA to incorporate those features to find better cognitive models. How do we know the cognitive model at hand needs to improve? Here is a simple two-bracket criteria – a cognitive model should perform better than the one skill model (one skill for all items) and the item model (each item being a skill). Researchers can run AFM on the cognitive model and compare cross validation errors with the two benchmark models. If the evaluation scores of the cognitive are worse than those of the benchmark cognitive models, it is worth improving. On the other side, if we are given student-item data without a cognitive model or our purpose is to do prediction or item recommendation, then we may use EPCA to do the initial factorization. If the result is better than the existing cognitive model, then construction of a P matrix and use of LFA may be warranted.

The quest for a better cognitive model should not stop here. On the technical side, there is a great potential for unsupervised learning method like EPCA. The out-of-sample prediction is likely to be better if the implementation of EPCA would allow extra constraints, thanks to a huge reduction of the parameters. This is entirely solvable. The second future direction is that to make LFA and EPCA more scalable to handle large data sets. As more and more student data are collected, it is not unusual to see data sets with more than 1000 students and 100 item responses for each student. The needs to handle such large data sets are increasing. Some of Singh's work is in this direction [16]. The third direction is to use a full Bayesian inference for both methods in the future. Such an approach should prevent over fitting.

8. Appendix

8.1 The derivation of the log likelihood function of AFM

AFM 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, $i = 1,...,n$

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

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

x_i, the value of the dependent variables for observation i

β**,** the parameters to estimate

Thus the log likelihood of the parameters given the data is

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

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

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

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

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

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

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

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

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

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

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

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

8.2 The derivation of the log likelihood function of CFM

CFM can be viewed as the Bernoulli case of a nonlinear generalized model with

$$z_{rk} = \theta_{ri} + \beta_{rk} + \gamma_{rk}T_{rk} \tag{17}$$

$$p_r = \prod_{k=1}^{K}(\frac{1}{1+e^{-z_{rk}}}) \tag{18}$$

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 Eq. (19).

$$l_{MLE} = \log Likelihood = \sum_{r=1}^{n}(y_r \log(p_r) + (1-y_r)\log(1-p_r)) \tag{19}$$

By taking the derivative of the log likelihood function with respect to the parameters, we derive the gradient for each parameter in Eq. (20), (21) and (22).

$$\begin{aligned}\frac{dl}{d\theta_i} &= \sum_{r=1}^{n}\frac{dl}{dp_r}\frac{dp_r}{d\theta_i} = \sum_{i=1}^{n}\left(\frac{p_r - y_r}{(p_r - 1)p_r}\cdot\sum_{k=1}^{K}\left(\frac{1}{1+e^{-z_{rk}}}\right)p_r\right) \\ &= \sum_{i=1}^{n}\left(\frac{p_r - y_r}{(p_r - 1)}\cdot\sum_{k=1}^{K}\left(\frac{1}{1+e^{-z_{rk}}}\right)\right)\end{aligned} \tag{20}$$

$$\frac{dl}{d\beta_k} = \sum_{r=1}^{n}\frac{dl}{dp_r}\frac{dp_r}{d\beta_k} = \sum_{i=1}^{n}\left(\frac{p_r - y_r}{(p_r - 1)p_r}\cdot\frac{e^{K\theta_i + \sum_{k=1}^{K}\beta_k + + \sum_{k=1}^{K}\gamma_k T_{kr}}}{\prod_{k=1}^{K}(1+e^{z_{rk}})}\cdot\frac{1}{1+e^{z_{rk}}}\right) \tag{21}$$

$$\frac{dl}{d\gamma_k} = \sum_{r=1}^{n}\frac{dl}{dp_r}\frac{dp_r}{d\gamma_k} = \sum_{i=1}^{n}\left(\frac{p_r - y_r}{(p_r - 1)p_r}\frac{T_{kr}e^{K\theta_i + \sum_{k=1}^{K}\beta_k + + \sum_{k=1}^{K}\gamma_k T_{kr}}}{\prod_{k=1}^{K}(1+e^{z_{rk}})}\cdot\frac{1}{1+e^{z_{rk}}}\right) \tag{22}$$

8.3 The Factors in the P matrix for the Geometry Data

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

8.4 The Factors in the P matrix for the EAPS Data

coverStory-bball
coverStory-donut
coverStory-lottery
coverStory-waiter
finalArith-div
finalArith-mult
numCategory-easy-bball
numCategory-easy-donut
numCategory-easy-lottery
numCategory-easy-waiter
numCategory-hard-bball
numCategory-hard-donut
numCategory-hard-lottery
numCategory-hard-waiter
numDifficulty-easy
numDifficulty-hard
origArith-div
origArith-mult
presentation-equation
presentation-story
presentation-word
unknownPosition-result
unknownPosition-start

8.5 Alternative Q matrices found by LFA search using CFM

Table 34 The lists of skills along with their parameter estimates from the best three Q matrices found by LFA search with penalized CFM on the EAPS data, ranked by BIC. The cross validation errors of the three models are listed in the last row.

Best Model	2nd Best	3rd Best
unknownPosition-start (0.82)	unknownPosition-start (0.94)	unknownPosition-start (0.89)
unknownPosition-result (2.52)	unknownPosition-result (3.38)	unknownPosition-result (2.8)
presentation-equation (0.65)	presentation-equation (0.67)	presentation-equation (0.63)
numDifficulty-hard (1.54)	numDifficulty-hard (1.58)	numDifficulty-hard (1.43)
origArith-div (1.73)	origArith-div (1.61)	origArith-div (1.75)
numCategory-hard-bball (0.85)	numCategory-hard-bball (0.84)	numCategory-hard-bball (0.83)
	finalArith-mult (2.34)	numCategory-easy-donut (2.08)
0.177	0.175	0.177

9. Bibliography

1. *Ants on the beach*. Available from: http://realtimecollisiondetection.net/blog/wp-content/uploads/2007/08/ant_beach.png.
2. Simon, H., *The Sciences of the Artificial*. 3rd ed. 1996: The MIT Press.
3. 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.
4. 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.
5. Sarkis, H., *Cognitive Tutor Algebra 1 Program Evaluation*. 2004, Available from Carnegie Learning, Inc., 1200 Penn Avenue, Suite 150, Pittsburgh, PA 15222.
6. 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.
7. Nathan, M.J. and A. Petrosino, *Expert Blind Spot Among Preservice Teachers.* American Educational Research Journal, 2003. **40**(4): p. 905–928.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. Koedinger, K. and M.J. Nathan, *The real story behind story problems:Effects of representation on quantitative reasoning.* Journal of the Learning Sciences, 2004. **13**(2): p. 129-164.
14. Russell, S. and P. Norvig, *Artificial Intelligence: A Modern Approach*. 2nd ed. 2003: Prentice Hall
15. Gordon, G., *Generalized2 Linear2 Models*, in *Annual Conference on Neural Information Processing Systems 2002*. 2002.
16. 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.

17. 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.
18. 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.
19. Embretson, S., *Multicomponent Response Models*, in *Handbook of Modern Item Response Theory* W.V.D. Linden and R.K. Hambleton, Editors. 1997, Springer.
20. von Davier, M., *A General Diagnostic Model Applied to Language Testing Data*. 2005, Educational Testing Service.
21. 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
22. 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.
23. Nichols, P., S. Chipman, and R. Brennan, *Cognitively diagnostic assessment*. 1995, Hillsdale , NJ: Erlbaum.
24. Davier, M.v., *A General Diagnostic Model Applied to Language Testing Data*. 2005, Educational Testing Service.
25. 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.
26. 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.
27. Ur, S. and K. VanLehn, *STEPS: A Simulated, Tutorable Physics Student.* Journal of Artificial Intelligence in Education, 1995. **6**(4): p. 405-437.
28. 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
29. 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.
30. Barnes, T., *The Q-matrix Method: Mining Student Response Data for Knowledge*, in *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*. 2005.
31. Draney, K., P. Pirolli, and M. Wilson, *A Measurement Model for a Complex Cognitive Skill*, in *Cognitively Diagnostic Assessment*. 1995, Erlbaum: Hillsdale, NJ
32. Pavlik, P.I., H. Cen, and K. Koedinger. *Performance Factors Analysis - A New Alternative to Knowledge Tracing* in *The 14th International Conference on Artificial Intelligence in Education* 2009.
33. Singliar, T. and M. Hauskrecht, *Noisy-or Component Analysis and its Application to Link Analysis.* Journal of Machine Learning Research, 2006. **7**: p. 2189-2213.

34. Harrell, F.E., *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. 2001: Springer.

35. Hastie, T., R. Tibshirani, and J.H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. 2009: Springer.

36. Cen, H., K. Koedinger, and J. B. *Comparing Two IRT Models for Conjunctive Skills*. in *the 9th International Conference on Intelligent Tutoring Systems*. 2008.

37. Wasserman, L., *All of Statistics: A Concise Course in Statistical Inference*. 2004: Springer

38. Rafferty, A. and M. Yudelson, *Applying Learning Factors Analysis to Build Stereotypic Student Models*, in *13th International Conference on Artificial Intelligence in Education (Best Student Paper)*. 2007: Los Angeles, CA.

39. Nwaigwe, A., et al., *Exploring alternative methods for error attribution in learning curves analyses in intelligent tutoring systems*, in *13th International Conference on Artificial Intelligence in Education* K.R.K. R. Luckin, J. Greer Editor. 2007, Amsterdam, Netherlands: IOS Press.: Los Angeles, CA.

40. Leszczenski, J.M., *Learning Factors Analysis Learns to Read (Thesis)*, in *Computer Science*. 2007, Carnegie Mellon University: Pittsburgh PA, 15213.

41. Leszczenski, J.M. and J.E. Beck, *What's in a Word? Extending Learning Factors Analysis to Model Reading Transfer*, in *13th International Conference on Artificial Intelligence in Education, Educational Data Mining Workshop*. 2007: Los Angeles, CA.

42. Cen, H., K. Koedinger, and B. Junker, *Is Over Practice Necessary? – Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining*, in *13th International Conference on Artificial Intelligence in Education*. 2007: Los Angeles, CA.

43. 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.

44. Pavlik, P., *Bridging the Bridge to Algebra: Measuring and optimizing the influence of prerequisite skills on a pre-algebra curriculum*. 2007.

45. 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.

46. Tipping, M.E. and C.M. Bishop, *Probabilistic Principal Component Analysis.* Journal of the Royal Statistical Society, 1999. **61**: p. 611–622.

47. Casella, G. and R. Berger, *Statistical Inference*. 2001: Duxbury Press.

MACHINE LEARNING
DEPARTMENT

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Carnegie Mellon