Review, computation and application of the Additive Factor Model (AFM)

Durand, Guillaume; Goutte, Cyril; Belacel, Nabil; Bouslimani, Yassine; Léger, Serge

https://doi.org/10.4224/23002483

Review, Computation and Application of the Additive Factor Model (AFM)

Guillaume Durand
National Research Council Canada
Guillaume.Durand@nrc-cnrc.gc.ca

Cyril Goutte
National Research Council Canada
Cyril.Goutte@nrc-cnrc.gc.ca

Nabil Belacel
National Research Council Canada
Nabil.Belacel@nrc-cnrc.gc.ca

Yassine Bouslimani
Université de Moncton
Yassine.Bouslimani@umoncton.ca

Serge Léger
National Research Council Canada
Serge.Leger@nrc-cnrc.gc.ca

October 20, 2017

Roughly a decade ago appeared the Additive Factor Model (AFM), a cognitive diagnostic model that was subsequently implemented by PSLC-Datashop and successfully used by researchers since then. While powerful, this model is not always simple to apprehend for a novice user. This paper aims at addressing this concern, by sharing our understanding of the model and the way we implemented it while conducting research on Q-matrices evaluation. Situating AFM as a member of the multidimensional Item Response Theory models, this paper provides readers with several considerations regarding the behavior of the model and the meaning of its parameters, as well as details on the technical implementation. Finally a use case is presented showing how the model can be used to monitor, understand and improve learning.

1. Introduction

The Additive Factor Model (AFM) is a cognitive diagnostic model proposed by Cen (Cen et al., 2006; Cen et al., 2008) in 2006 and implemented in the Pittsburgh Science of Learning Centre (PSLC) Datashop website (Koedinger et al., 2010). Closely related to Item Response Theory (IRT), AFM calculates learners success probabilities using user and skill specific parameters. AFM is a multidimensional model in which success relies not only on a single but potentially on multiple abilities. These abilities are defined in a parameter that associates items to skills in a Q-matrix or Knowledge component model,[1] as it is called in PSLC-Datashop (Koedinger et al., 2010). It is possible to trace the Q-matrix concept back to the Rule-Space Model introduced in the eighties (Tatsuoka, 1983) to statistically classify students item responses into a set of ideal response patterns associated to different cognitive skills. More precisely, Rule-Space aimed at performing fine granularity diagnostic assessments considering an incidence matrix populated

[1] In this paper, the authors use the words “skill”, “knowledge component” and “competency” indistinctly. This is also the case for the words “Q-matrix”, “assessment map” and “knowledge component model”.

by the potential pool of items representing all $(2^K - 1)$ combinations of K skills that an item could evaluate (Birenbaum et al., 1992). Using a “skill hierarchy”, Rule-Space proposed to reduce and simplify the incidence matrix into a binary Q-matrix, as illustrated in Figure 1. Q-matrices have been of interest for cognitive diagnostic modeling but also for researchers in the Intelligent Tutoring Systems (ITS) and Educational Data Mining (EDM) communities for years (Barnes, 2005; Desmarais, 2011b; Sun et al., 2014; Liu et al., 2012; Stamper and Koedinger, 2011). Having an accurate mapping of the skills involved in solving items provides more reliable assessment metrics allowing learning systems to draw inferences and diagnostics and to provide accurate intelligence. For this purpose Q-matrices evaluation has been conducted through the lens of AFM and other diagnostic models.

$$\text{Q} = \begin{array}{c} \\ Item1 \\ Item2 \\ Item3 \\ Item4 \end{array} \begin{array}{c} \begin{array}{cccc} Skill1 & Skill2 & Skill3 & Skill4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

Figure 1: Example of a Q-matrix with one skill per item for the first three items and two skills for the fourth one.

For instance, in the cognitive diagnostic modeling community, Rupp and Templin (Rupp and Templin, 2008) evaluated the impact of ill-defined Q-matrices on the Deterministic Inputs, Noisy “And” gate (DINA) model (Junker and Sijtsma, 2001) using a simulation dataset. Considering 4 attributes and 15 items, they generated 10,000 respondent answers using a normal distribution of skill patterns. They defined Q-Matrix misspecification conditions by eliminating blocks of items (adding or removing skills for specific items) and representing incorrect dependency relationships (adding or removing skills when specific skills are present or missing). Under the effect of misconceptions, and depending on the category of misconceptions, the estimated guess and slip values became larger than the initial ones. These results were also observed and confirmed by De la Torre (De la Torre, 2008) in his δ-method.

In the EDM community, Q-matrices have been of interest since early work by Barnes (Barnes, 2005), or in the study of matrix factorization techniques by Desmarais and others (Desmarais, 2011a; Desmarais, 2011b; Desmarais and Naceur, 2013; Durand et al., 2015) but also through AFM as proposed in PSLC-Datashop (Stamper and Koedinger, 2011). In PSLC-Datashop, a Q-matrix can be uploaded in an on-line system, and evaluated on reference datasets available in the platform, in order to provide several goodness of fit metrics as well as tools to quantitatively analyze each skill mapping. The PSLC-Datashop is an outstanding tool to evaluate Q-matrices and perform Learning Factor Analysis (Cen et al., 2006). However, contrary to, for example, DINA (De la Torre, 2009), we were not able to find in the literature any source code or hands-on parameter estimation primer allowing users to build an AFM evaluator to iteratively refine Q-matrices. Using PSLC-Datashop for this purpose would be cumbersome as all modeling is done on the PSLC server on which both dataset and Q-matrix have to be uploaded. In order to address this need, we had to build our own implementation. This paper attempts to give the reader an understanding of the AFM model and how to implement it from easily available components, such as a first order unconstrained optimizer and a scientific computation language like Matlab.

In the following, we first provide a very selective overview of IRT to introduce AFM as a specific member of the multidimensional item response theory (MIRT) family before going deeper into the mathematics of the model and the parameters estimation. Finally, we describe the implementation we made as well as a use case on a Canadian engineering curriculum.

2. Item Response Theory and AFM

It would be very ambitious and highly controversial to write an exhaustive survey of IRT research in this report. However, any researcher willing to use AFM will have to answer the question of why using this model and not another one, likely known as an IRT model. As a result, we decided to focus on the key elements that could justify the usage of AFM, making a highly selective literature review omitting many points that IRT researchers would consider important to deeply understanding IRT models.

Many IRT researchers (Bock, 1997; Sijtsma and Junker, 2006) usually trace IRT back to the beginning of the 20th century with standardized intelligence tests and the preliminary concepts and methodologies of classic test theory (CTT). CTT was a very popular approach in psychometrics up to the end of the nineteen sixties. Statistical CTT models express an observable test score based on examinee performance and random error. The purpose of CTT is to determine how the score is influenced by the error. This is particularly useful in standardized tests to detect and correct potential biases that would affect examinees with unexpected result scores. These biases could be related, for instance, (but not limited) to cultural, gender, racial factors or unintended skills that are unfortunately not always obvious to characterize even for experts in the field. For example, a test evaluating leadership could wrongly attribute weak leadership to girls from a culture privileging men as leaders. In such a test, CTT would detect an unexpected error distribution, helping test designers to address the issue. IRT goes beyond CTT, and the transition from CTT to IRT occurred relatively naturally. Bock (Bock, 1997) identifies a paper by Thurstone (Thurstone, 1925) as the origin of IRT. Thurstone proposed to express the probability of success on a specific item as a function of a respondent's attribute. In his proposal, Thurstone made a bridge between CTT and what became IRT, using a random error as in CTT, and a probabilistic function to model the distribution of successes as in IRT. In that specific case the model was based on a standard, Normal distributed error. However, although Thurstone used a measurable variable, the "age of the respondents", modern IRT models are usually based on latent variables that are not directly observable. From Thurstone, researchers have been looking for better models explaining responses given by examinees and providing a better fit to observations of the Item Response Function (IRF).[2] Lord (Lord, 1952) was the first to see in different IRF a cumulative normal function and introduced two latent parameters: the examinee proficiency θ (later called α[3] in the Educational Data Mining literature and this paper) and the difficulty of the item b (a.k.a. β here). Further developments provided the IRT community with one of the most well-known and fundamental model, the Rasch model and a variant called the one parameter logistic model (1PL model), approximating Lord's cumulative normal distribution by a logistic function (Birnbaum, 1968) given by:

$$P(Y_{ij} = 1 | \alpha_i, \beta_j) = \frac{\exp(\alpha_i - \beta_j)}{1 + \exp(\alpha_i - \beta_j)} = f\left(\alpha_i - \beta_j\right) \tag{1}$$

[2] The IRF is the function that models the response of a person to an item.

[3] This parameter is not related to Cronbach's-α.

where $f(x) = e^x/(1+e^x) = 1/(1+e^{-x})$ is the logistic function, which transforms a real value into a probability in $]0;1[$. Y_{ij} is the response of student i on item j, with the convention that $Y_{ij} = 1$ for success and $Y_{ij} = 0$ for failure.[4] Parameter α_i is the ability or proficiency of student i, and β_j is the difficulty of item j . With this simple model, the more proficient the student becomes, the higher her probability of success. Conversely, the more difficult an item is, the lower her probability of success on this item. The Rasch model relies on three key assumptions (Sijtsma and Junker, 2006):

1. Local, or conditional, independence (LI): events are independent from each others. The result on one item does not depend on results obtained on previous ones; In other words, the past has no influence on the current probability of success.

2. Monotonicity: The item response function is monotonous; in other words, the probability of success increases when ability improves.

3. Unidimensionality: Attributes of the models are one dimensional; an examinee has the same proficiency on each item, and an item has the same difficulty for each learner.

Local independence is a strong assumption. Applied to cognitive tests, it rules out, for example, response patterns between items. If the correct answer is always the second choice, a student could learn that pattern quickly and apply it to the next item. In addition, the LI assumption typically does not hold when multiple attempts on items are possible. As a consequence, IRT needs to be used carefully, for example considering only the first attempt on each item in order to minimize the risk that the LI assumption does not hold. The LI assumption also requires to limit the formative feedback in order to prevent learning while testing. While monotonicity seems well accepted, the unidimensionality assumption may limit the scope of the tests that can be conducted. For example, let's consider an item and two examinees with the same proficiency: an IRT model complying with the unidimensionality assumption would predict identical probability of success. However, if proficiency is theoretically seen as an inherent capability of the examinee to solve the items considered, in practice, two items with the same level of difficulty may require different skills. Similarly, examinees with the same proficiency may master skills differently, suggesting different probabilities of success on the same item. For instance, let's assume that an item solving an addition and an item solving a subtraction have the same difficulty. They still require different skills. Two students having the same proficiency may master these skills differently, leading to different probability of success. IRT cannot provide an accurate model in that case.

MIRT (Multidimensional IRT) models address these situations by relaxing the unidimensional assumption and transforming single attributes to sets of attributes. This transformation comes with different ways of combining attributes to model examinee's success. Depending on the MIRT model considered, proficiency may transform into a set of proficiencies or skills and their interaction or combination towards examinee's success can be additive, conjunctive or disjunctive. A *compensatory* or *additive* model assumes that each skill adds to the item success. Compensatory models are useful to evaluate the sensitivity of student responses to the skills associated to the items (Sijtsma and Junker, 2006). A *conjunctive* model assumes that *all* skills in an item are necessary for success. The *disjunctive* model assumes that knowledge of *any* skill

[4] The opposite convention works just as well, reversing the sign of the parameters.

in the set yields success. A good example of compensatory MIRT model has been proposed by Reckase (Reckase, 1997) (multidimensional 3PLM).

$$\begin{aligned} P(Y_{ij}=1|\alpha_i, a_j, \beta_j, c_j) &= c_j + (1-c_j)\frac{\exp(a_{j1}\alpha_{i1} + ... + a_{jk}\alpha_{ik} - \beta_j)}{1+\exp(a_{j1}\alpha_{i1} + ... + a_{jk}\alpha_{ik} - \beta_j)} \\ &= c_j + (1-c_j)f(a_{j1}\alpha_{i1} + ... + a_{jk}\alpha_{ik} - \beta_j) \end{aligned} \quad (2)$$

Reckase reused the logistic cumulative distribution similar to the Rasch model and added c that is a base probability of success when the α abilities are very low; it can be seen as a pseudo-chance level or a guessing parameter. In this model a controls the slope of the item response function, and α becomes a *vector* of student abilities while β models the item difficulty .

By constraining some parameters of Reckase's model, Li (Li et al., 2012) defined a simpler multidimensional Rasch model modeling more explicitly the impact of the additive combination of abilities on the probability of success, again with a single item difficulty parameter:

$$P(Y_{ij}=1|\alpha_i, \beta_j) = \frac{1}{1+\exp[-(\alpha_{i1}+\alpha_{i2}....+\alpha_{ik}-\beta_j)]} = f(\alpha_{i1}+\alpha_{i2}....+\alpha_{ik}-\beta_j) \quad (3)$$

where $\alpha_{i1}+\alpha_{i2}....+\alpha_{ik}$ models the cumulative impact of the k abilities of examinee i on the probability of success. The Reckase or Li models define skills as a multidimensional examinee proficiency. However, instead of splitting the student proficiency into skills as in Eq. 2-3, a multidimensional model could also split the item difficulty (β) into multiple parameters related to the skills involved. Transforming β into multidimensional parameters change the focus of the model since the objective is not primarily to explore the learner's abilities, but rather to study the relationship between items and skills as represented in a Q-Matrix. One of the key feature of AFM (Cen et al., 2007; Cen et al., 2008) that we describe in the following section, is its unique decomposition of β into multidimensional skill related sub-parameters.

3. THE ADDITIVE FACTOR MODEL

Compared to the Rasch model and its student-based multidimensional derivatives, AFM (Cen et al., 2007; Cen et al., 2008), through its decomposition of β, takes into account the fact that an item may cover a combination of skills that have been addressed by other items. The estimated probability of success may also increase with repeated exposures to the same skill. AFM improves the modeling of student success on these two aspects as follows:

1. It relies on a Q-matrix (Tatsuoka, 1984) that represents the mapping from items to skills, and

2. It takes into account repeated exposures to a skill through a learning rate.

Given a group of I students and a set of J items, AFM expresses the probability that a student $i \in \{1 \ldots I\}$ succeeds on an item $j \in \{1 \ldots J\}$ by a mixed-effect logistic regression. The probability of success is expressed as:

$$\begin{aligned} P(Y_{ij}=1|\alpha_i, \boldsymbol{\beta}, \boldsymbol{\gamma}) &= \frac{\exp\left(\alpha_i + \sum_{k=1}^{K}\beta_k q_{jk} + \sum_{k=1}^{K}\gamma_k q_{jk} t_{ik}\right)}{1+\exp\left(\alpha_i + \sum_{k=1}^{K}\beta_k q_{jk} + \sum_{k=1}^{K}\gamma_k q_{jk} t_{ik}\right)} \\ &= f\left(\alpha_i + \sum_{k=1}^{K}\beta_k q_{jk} + \sum_{k=1}^{K}\gamma_k q_{jk} t_{ik}\right) \end{aligned} \quad (4)$$

with $f(x) = 1/(1 + e^{-x})$ the standard logistic function, as before, and:

Y_{ij} is the binary response of student i on item j;

α_i is the proficiency of student i;

β_k is the easiness of skill $k \in \{1 \ldots K\}$;[5]

γ_k is the learning rate for skill $k \in \{1 \ldots K\}$;

q_{jk} is a binary indicator that item j uses skill k, contained in the $J \times K$ Q-matrix $\mathbf{Q}$;

t_{ik} is the number of times student i has practiced skill k, a.k.a. opportunity;

K is the total number of skills (number of columns in the Q-matrix (Barnes, 2005)).

The extension of the Rasch model into AFM becomes apparent with a specific model configuration. If we take $\mathbf{Q}$ as a diagonal matrix with one skill per item (Figure 1), we have $q_{jk} = 1$ iff $j = k$ and 0 otherwise, so the first sum over k in Equation 4 reduces to β_j. Discarding the effect of learning by setting the learning rates to $\gamma_k = 0$, the AFM model reduces to $P(Y_{ij}|\alpha_i, \boldsymbol{\beta}) = f(\alpha_i + \beta_j)$ which is identical to the Rasch model (Eq. 1), except that β_j models item easiness rather than difficulty.

The interpretation of how the AFM model works in relation to the three parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ is that the probability of success Y_{ij} depends in a fairly straightforward manner on:

- Natural proficiency of student i, α_i: the higher the proficiency, the more likely the success, on all items.

- Easiness of skill k, β_k: the easiest a skill is, the more likely the success on items that contain that skill, for all students.

- Learning rate for skill k, γ_k: the more often a student practices an item with skill k, the higher the probability of success on future items with skill k.

In summary, the probability of success is influenced by α_i for a student independently of items or skills, one or several β_k, depending on the item, independently of students, while γ models the impact of the sequence of presentation of items.

In a practical situation, we are given an items-to-skills mapping $\mathbf{Q}$ and we observe the success or failure of students on a number of items, Y_{ij}, from which we compute the opportunity t_{ik}. Note that t_{ik} do not form a matrix, they are actually counters of how many times student i has been exposed to skill k. The value of t_{ik} may therefore increase, for a student i, as he meets new items with skill k. Estimating Rasch or AFM parameters can be done by maximizing the joint or conditional likelihood using, for example, efficient first-order numerical optimization methods on the likelihood function and its derivatives, as detailed in the following section.

[5] Note that the sign of β_k in Eq. 4 is opposite that of β_j in Eq. 1, as β_k models easiness rather than difficulty.

3.1. Parameters Estimation

Given an item-to-skills model $\mathbf{Q} = [q_{jk}]$, observations $\mathbf{Y} = [y_{ij}]$, and a sequence of opportunities t_{ik}, the AFM parameters we need to estimate are the α_i, β_k and γ_k.

Let us introduce

$$z_{ij} = \left(\alpha_i + \sum_{k=1}^{K} \beta_k q_{jk} + \sum_{k=1}^{K} \gamma_k q_{jk} t_{ik} \right)$$

and rewrite the model as

$$P(Y_{ij} = 1 | \alpha_i, \boldsymbol{\beta}, \boldsymbol{\gamma}) = f(z_{ij}) = \frac{1}{1 + \exp(-z_{ij})} = p_{ij}.$$

Cen (Cen, 2009) proposes a re-parameterization of z_{ij} as $z = \boldsymbol{\theta}^T . \boldsymbol{x}$, for a suitable definition of $\boldsymbol{\theta}$ involving $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, and of $\boldsymbol{x}$ involving q_{jk} and t_{ik}. However, this is not necessary if we simply view parameter estimation as a generic optimization problem.

Parameters are fit by maximizing a penalized likelihood (Cen et al., 2008). The likelihood is the probability of the observations, given the model parameter. The observations about student success is a binary observation $y_{ij} \in \{0; 1\}$ such that $y_{ji} = 1$ if student j has succeeded item i and $y_{ij} = 0$ otherwise. The probability of this observation given by our model is $P(Y_{ij} = 1 | \alpha_i, \boldsymbol{\beta}, \boldsymbol{\gamma})$ if $y_{ij} = 1$ and $(1 - P(Y_{ij} = 1 | \alpha_i, \boldsymbol{\beta}, \boldsymbol{\gamma}))$ otherwise. The likelihood for observation y_{ij} can therefore be written as $p_{ij}^{y_{ij}} (1 - p_{ij})^{1 - y_{ij}}$. Going to the log domain and mimicking Section 8.1 of Cen's thesis (Cen, 2009):

$$\begin{aligned} \log \left(p_{ij}^{y_{ij}} (1 - p_{ij})^{1 - y_{ij}} \right) &= y_{ij} \log p_{ij} + (1 - y_{ij}) \log(1 - p_{ij}) \\ &= y_{ij} \log \left(\frac{e^{z_{ij}}}{1 + e^{z_{ij}}} \right) + (1 - y_{ij}) \log \left(\frac{1}{1 + e^{zij}} \right) \\ &= y_{ij} z_{ij} - y_{ij} \log(1 + e^{z_{ij}}) - (1 - y_{ij}) \log(1 + e^{z_{ij}}) \\ &= y_{ij} z_{ij} - \log(1 + e^{z_{ij}}) \qquad (5) \end{aligned}$$

Assuming independent observations, the likelihood for all observations is the product of individual observation likelihood, so that in the log domain, the log-likelihood is the sum of Eq. 5 over all observations:

$$\mathcal{L} = \sum_{ij} y_{ij} z_{ij} - \log(1 + e^{z_{ij}}) \qquad (6)$$

Maximizing the likelihood directly over many parameters is prone to over fitting, resulting in unreasonably high (or low) estimates for some parameters. As a consequence, it is usually better to use a penalized version, which induces an overwhelming cost to large parameter values, controlled by a hyper parameter λ. This will be discussed in more details in Section 3.3. The cost to maximize becomes:

$$\mathcal{C}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \sum_{ij} y_{ij} z_{ij} - \log(1 + e^{z_{ij}}) - \frac{\lambda}{2} \left(\sum_i \alpha_i^2 \right) \qquad (7)$$

As noted by Cen (Cen, 2009), this is equivalent to maximizing the posterior probability of the parameters given the observations, with a Gaussian prior on proficiency parameters. Estimating the parameters is therefore an unconstrained optimization problem: maximizing $\mathcal{C}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$

with respect to $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. Generic multidimensional optimization problems such as conjugate gradient or quasi-Newton (Fletcher, 1987) work by repeatedly choosing a direction in parameter spaces, and doing a one-dimensional optimization (line search) in that direction, until convergence conditions are met. A good strategy is to use a conjugate gradient algorithm with approximate line search, such as implemented in (Rasmussen, 2006). The only information needed to perform such an optimization is a function that calculates, for each set of parameters, the value of the cost (Eq. 7) and the derivatives of that cost with respect to each parameter, which we cover below.

3.1.1. Computing Partial Derivatives

The derivatives of the penalized cost w.r.t. the parameters are straightforward:

$$\frac{\partial C}{\partial \alpha_i} = \frac{\partial \mathcal{L}}{\partial \alpha_i} - \lambda \alpha_i, \quad \frac{\partial C}{\partial \beta_k} = \frac{\partial \mathcal{L}}{\partial \beta_k} \quad \text{and} \quad \frac{\partial C}{\partial \gamma_k} = \frac{\partial \mathcal{L}}{\partial \gamma_k} \tag{8}$$

The derivatives of the log-likelihood (Eq. 6) are obtained by summing over the derivatives of individual observation likelihood. In our notation, the z_{ij} are functions of the parameters, and

$$\frac{\partial \mathcal{L}}{\partial \alpha_i} = \sum_j y_{ij} \frac{\partial z_{ij}}{\partial \alpha_i} - \frac{1}{1+e^{z_{ij}}} \frac{\partial (1+e^{z_{ij}})}{\partial \alpha_i} = \sum_j (y_{ij} - \frac{e^{z_{ij}}}{1+e^{z_{ij}}}) \frac{\partial z_{ij}}{\partial \alpha_i} = \sum_j (y_{ij} - p_{ij}) \frac{\partial z_{ij}}{\partial \alpha_i}, \tag{9}$$

because $\partial z_{i'j} / \partial \alpha_i = 0$ for all $i' \neq i$. Similarly, for derivatives w.r.t. parameters β_k and γ_k we get from (Eq. 6):

$$\frac{\partial \mathcal{L}}{\partial \beta_k} = \sum_{ij} (y_{ij} - p_{ij}) \frac{\partial z_{ij}}{\partial \beta_k}, \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \gamma_k} = \sum_{ij} (y_{ij} - p_{ij}) \frac{\partial z_{ij}}{\partial \gamma_k}. \tag{10}$$

Notice the straightforward interpretation for these derivatives: the further the observation y_{ij} is to the probability assigned by the model p_{ij}, the larger its contribution to the derivative.

The derivatives of z_{ij} are straightforward as z_{ij} is just a weighted sum:

$$\frac{\partial z_{ij}}{\partial \alpha_i} = 1, \quad \frac{\partial z_{ij}}{\partial \beta_k} = q_{jk} \quad \text{and} \quad \frac{\partial z_{ij}}{\partial \gamma_k} = q_{jk} t_{ik} \tag{11}$$

In order to keep the learning rate γ_k positive (Section 3.3.), we can parameterize it as $\gamma_k = (g_k)^2$ (or similarly $\gamma_k = \exp g_k$), forcing γ_k to be positive for all values of the new parameter g_k. The calculation of partial derivatives w.r.t g_k are similar until:

$$\frac{\partial z_{ij}}{\partial g_k} = 2 g_k q_{jk} t_{ik} \tag{12}$$

Equipped with a generic multidimensional optimization algorithm and the above derivatives, we can now optimize $\mathcal{C}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ with respect to $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. Other, related models may be optimized in a similar way, as long as we can compute derivatives of a cost function (such as the penalized log-likelihood) and plug them into the optimization routine.

3.1.2. Understanding Derivatives

Let's pause for a second to look at the derivatives in order to understand what the minimization is doing to the model. When we optimize, we want the derivatives of the cost to be 0. According to Equations 9 and 10, the overall derivative is a sum of contributions from each example y_{ij}. Each contribution is the product of two terms: 1) the derivative of z_{ij} w.r.t. the parameter and 2) a term that is the difference between the observation y_{ij} and the model estimate p_{ij}. The derivative w.r.t. student proficiency α_i is always 1 for the parameter corresponding to student i, and 0 for the other α parameters. Accordingly, the derivative has no influence on parameters corresponding to other students, and for α_i, it is trying to make the model estimate as close as possible to the observations, or at least make estimation errors cancel out across the data. For the skill easiness β_k, when the skill is not connected to the observed item, $q_{jk} = 0$ and the derivative is 0. In other words, an observation has no influence on parameters related to skills it is not connected to, which makes sense. On the other hand, if $q_{jk} = 1$, the derivative is trying to set β such that estimation errors are minimized or cancel out. Similar reasoning applies to γ_k, except that in this case, the influence of the estimation error is magnified by the number of times the skill has been tried: it will be more important to minimize estimation error on observations connected to skills that have been tried a lot. The impact of the penalization term is to always pull α towards 0, pulling stronger when α is larger.

3.2. Learning Curves

The learning curve *for a student i and a skill k* shows how student i gets better with repeated exposure to that skill. This is given by the equation:

$$LC_{ik}(t) = f\left(\alpha_i + \beta_k + \gamma_k t\right) \tag{13}$$

This is the AFM equation, Eq. 13, for student i and an imaginary item having a unique skill k. It estimates the probability that student j succeeds on an item with skill k as unique skill. It may not be straightforward to relate to observed student performance on items that mix different skills. A learning curve *for skill k* may be obtained by replacing the student-specific parameter by the average student proficiency, or more conveniently by zero. This also results in a logistic curve, and may also be difficult to relate to observed, individual student performance.

One strategy for relating a learning curve (Eq. 13) to observed data is to average performance across all students, at each opportunity. This is the approach adopted in the PSLC-Datashop (Koedinger et al., 2010). One issue is that in general, not all students have taken the same number of items, or taken the items in the same order (which impacts the per-skill opportunity). Another issue is that there may be be a bias in the averaged observed performance as the number of opportunities grow. For example if low-performing students are encouraged to take more item for a skill they fail on, average performance will tend to under-shoot the model estimates.

Another way of tracing a learning curve (Ritter and Schooler, 2001) is to rely on success rates (or error rates) aggregated without using AFM parameters. In this case, the resulting curve can help detect interesting patterns. For example, when two skills are incorrectly modeled as one the empirical learning curve may display two spikes before smoothly reaching its success rate asymptote (Corbett and Anderson, 1994). Also, while it is easy to trace empirical learning curves when each item is associated to one skill, it is more challenging when items are associated with multiple skills. In this case the compensatory mechanisms between skills may not be obvious empirically. AFM parameters provide precious indication on the relationships between the multiple skills involved in order to explain observed results.

The quality of the resulting AFM learning curves and their interpretation is strongly linked to the amount of data available for estimating them. This suggests a number of practical recommendations:

- Students: Each skill should have been tried by a minimum number of students, so that AFM parameters are reliable. PSLC-Datashop puts the default threshold to 10 students.This means that each opportunity iteration should have been met by at least 10 students.

- Opportunity: As learning curves measure how performance evolves when opportunities increase, it is also essential that skills have been met by some users several times. The default threshold used by PSLC-Datashop is 2 opportunities. A larger number would be advisable, however, as the shape of the learning curve depends on the number of points used to build it. A minimum of five opportunities reached should be necessary to interpret the learning rate.

- Error rate: If the learning curve remains consistently high or low, the learning rate can be questionable. PSLC-Datashop considers a low error threshold to be under 20% failure (above 80% success) and a high error threshold to be above 40% failure (under 60% success). We will see in Section 4.2. that extreme success rates can also be the consequence of AFM having difficulties to discriminate properly the compensation effects between skills. We will talk about compensation effects in Section 4..

3.3. Constraints on AFM parameters computation

Values of the AFM parameters can be used to evaluate qualitatively a knowledge component model. When the number of observations is low for some students, items or skills, the estimation may yield poorly identified parameters and unreliable estimates. In order to avoid model degeneration and to accelerate its computation, we consider a number of constraints on the parameters.

α_i (natural proficiency of student i): Birnbaum (Birnbaum, 1968) showed that the latent ability α_i follows a Normal distribution. This means that values of α outside $[-3;+3]$ are unlikely. This motivates the use of the penalty on α introduced earlier (Eq. 7) as well as a way to tune it: When values of α fall outside of the target interval, we increase λ until they fit inside. Penalizing α also improves computational efficiency as optimizing the penalized cost takes fewer iterations.

β_k (easyness of skill k): The easiness is related to the intercept of the learning curve. Although there doesn't seem to be an expected range for β, too high or too low values will put the model into the flat parts of the logistic function, where no learning occurs. A very negative value of β_k can model a very hard skill requiring either an overwhelming number of opportunities, or a correspondingly large learning rate in order to yield probabilities significantly above zero. Although neither PSLC nor our implementation penalize β, it could make sense to do so to improve computational efficiency.

γ_k (learning rate for skill k): The learning rate is closely related to the slope of the learning curve. Previous works suggest that the learning curve should follow a power law (Newell

Algorithm 1: Pseudo-code of the main parameter estimation function. Values in square brackets are defaults.

```
Data: Q-matrix $Q$ and transactions $T$
Parameters: Penalization parameter $\lambda$ [0.01], maximum number of line search $n$ [100]
Result: $\alpha$, $\beta$, $\gamma$
Set initial parameters $\alpha_0, \beta_0, g_0$ [0,0,0] ;
Calculate opportunity at each transaction for each skill ;
Maximize Eq. 7 w.r.t. parameters $\alpha, \beta, \gamma$ using unconstrained optimization:
repeat
    Pick search direction;
    Minimize Eq. 7 along search direction (line search);
until $n$ line search, or tolerance reached;
```

and Rosenbloom, 1981) or that an exponential function may be a better fit in some circumstances (Murre and Chessa, 2011). For AFM, the main practical constraint is that the learning rate γ_k must be positive (as a negative value would imply that the skill is being unlearned). This is implemented using the re-parameterization described in Section 3.1.1..

4. Implementation and use case

4.1. Implementation

The estimation of AFM parameters can be carried out using a general unconstrained optimization technique as presented in Algorithm 1. The main function takes as input a set of *transactions* containing the assessment results for each learner, as well as the mapping between assessment and skills (Q-matrix). From this information we compute the number of opportunity, i.e. the number of times each learner met each skill at her first attempt of the assessment. Once parameters are set at their initial values (which default to zero), we can run the unconstrained optimization routine to optimize the regularized cost in Equation 7 with respect to α, β and γ.

Optimization is typically run by picking a direction in parameter space and optimizing the cost in one dimension along that direction, a.k.a. a *line search*. Optimizing each parameter in turn or picking the direction of steepest descent usually leads to slow convergence. Better techniques are the conjugate gradient or the quasi-Newton algorithm, where the gradient of the cost (indicating the steepest descent) is combined with the previous search direction to produce a new search direction. This produces much quicker convergence towards a local minimum. In practice, optimization ends when a pre-specified number of line search have been run, or progress between line search has stopped (within a small tolerance). Implementations of general unconstrained optimization are widely and freely available in various language.[6]

Additional parameters controlling the estimation are:

- The hyper-parameter λ weighting the trade-off between the regularizer and the likelihood in the cost function (Eq. 7), in order to keep α within meaningful boundaries;

- The maximum number of line search n, which allows the user to limit the runtime.

[6] In our Octave/Matlab implementation we use the conjugate gradient implementation of (Rasmussen, 2006).

```
Algorithm 2: Regularized cost and its partial derivatives w.r.t. parameters.
    Data: Q-matrix: $Q = [q_{jk}]$ and transactions: $T = [i, j, y_{ij}, t_{ik}]$
    Input: Parameters $[\alpha; \beta; \gamma]$ and regularization parameter $\lambda$
    Output: Cost $C(\alpha, \beta, \gamma)$ and derivatives $\frac{\partial C}{\partial \alpha}(\alpha, \beta, \gamma)$, $\frac{\partial C}{\partial \beta}(\alpha, \beta, \gamma)$ and $\frac{\partial C}{\partial \gamma}(\alpha, \beta, \gamma)$
    Compute $z_{ij} = \alpha_i + \sum_{k=1}^{K} \beta_k q_{jk} + \sum_{k=1}^{K} g_k^2 q_{jk} t_{ik}$;
    Compute $C(\alpha, \beta, \gamma)$ according to Eq. 7;
    Set partial derivatives of $z_{ij}$ according to Eq. 11;
    Compute partial derivatives of log-likelihood according to Eq. 9-10;
    Compute partial derivatives of the regularized cost according to Eq. 8.
```

At the heart of the optimization routine in Algorithm 1 is a function that computes the regularized cost and its derivatives with respect to the parameters, as described in Algorithm 2. The first step is to compute $z_{ij} = \alpha_i + \sum_{k=1}^{K} \beta_k q_{jk} + \sum_{k=1}^{K} \gamma_k q_{jk} t_{ik}$ for all pairs of students and items observed in the dataset. This allows to compute the value of the regularized cost according to Eq. 7. In order to compute the derivatives, we first set the partial derivatives of z_{ij} w.r.t. parameters using Eq. 11, then compute the derivatives of the log-likelihood using Eqs. 9 and 10. Note that this can be done efficiently using a single vector-matrix multiplications by building a vector containing the estimation error $(y_{ij} - p_{ij})$ for all pairs of observed (i, j), and a large matrix indexing all observed pairs of (i, j) in rows, and all parameters in column, with each cell containing the partial derivative of z_{ij} w.r.t. to the parameter of the corresponding column, set according to Eq. 11.[7] The derivative of the log-likelihood is the product of these. The derivative of the regularization term is then simply added according to Eq. 8.

In our Matlab implementation[8], all parameters are stored in a single vector containing α in the first I slots, β in the next K slots and γ in the final K slot. The cost returned by Algorithm 2 is a scalar, and the partial derivatives w.r.t. parameters is a vector indexed similarly as the parameter vector, with $\partial C / \partial \alpha$ in the first I slots, etc. Most of the computational effort in the parameter estimation is spent calculating values and derivative of the regularized cost. The main source of improvement for speeding up the estimation is therefore optimizing this computation, using efficient sparse multiplication operations between vectors and matrices.

4.2. Application example

In this section we propose to apply AFM parameters estimation and learning curves analysis on data obtained from an engineering curriculum. This is unnatural territory for AFM, a model ordinarily trained on data coming from Intelligent Tutoring Systems. In spite of this, we want to show how it can be exploited in this other educational area that, as we believe, could bring benefits for a wider audience. However, we present here a limited example that aims mainly at showing the capacities of the models and introducing the reader to the techniques to use it. For the audience interested in a more detailed and more usual use case a recent example using PSLC-Datashop functionalities can be found in (Rivers et al., 2016).

As mentioned earlier in Section 2., AFM is a cognitive diagnostic model that predicts student success based on past tests performances and their associated assessment map (Q-matrix). Similarly to many machine learning models, AFM needs to learn its parameters from the past

[7] This matrix is sparse as many q_{jk} and all $\partial z_{ij} / \partial \alpha_l, l \neq i$ are zero.

[8] This implementation is available on request by contacting first authors of this document.

to predict the future. In this exercise, data quality has a strong impact on the predictions made and it is essential to gain as much data understanding before running any analysis and making inferences. In the following, we introduce the data set as well as its pre-processing and analysis.

4.2.1. Data set

The example data set contains 356 observations obtained from 17 student transcripts listing scores from 0 (Failure) to 4.3 (Perfect) in 24 different courses completed from fall 2009 to winter 2015 in a Canadian University. The underlying assessment map is derived from 12 graduate *attributes* defined by the Canadian Engineering Accreditation Board (CEAB) (Canadian Engineering Accreditation Board, 2014). These attributes are used by CEAB as criteria to evaluate Canadian engineering programs for the purpose of accreditation. For each attribute the engineering faculty defined between 2 and 6 skills and associated them to the different courses of the program (Table 1).

Table 1: The 12 graduate attributes used by the Canadian Engineering Accreditation Board and the number of skills associated by the engineering faculty.

Label	Skills	Name
QR01	4	A knowledge base for engineering
QR02	4	Problem analysis
QR03	4	Investigation
QR04	5	Design
QR05	4	Use of engineering tools
QR06	4	Individual and team work
QR07	6	Communication skills
QR08	3	Professionalism
QR09	2	Impact of engineering on society and the environment
QR010	2	Ethics and equity
QR011	3	Economics and project management
QR012	2	Life-long learning

The resulting Q-matrix (Figure 2) comprises a total of 43 skills associated to the 24 different courses of an electrical engineering program. Considering constraints listed in Section 3.2., we can notice that the number of skills seems relatively important in comparison to the number of courses and observations. As a result, some skills are associated with low opportunity numbers, preventing the user to make inferences on them. Fortunately, not all skills are in this situation. On average each student met each skill 2.9 (s= .93) times.

4.2.2. Data pre-processing

Scores have to be above zero for the student to pass the course. With this definition of success and failure, we obtained only 16 failures (out of 356 records) for 7 different students and 8 courses. This number is relatively low and limits the number of interesting skills as most would have a flat learning curve reflecting the perfect success rate on the corresponding courses.

A way to improve the dataset using the scores is to raise the success criteria in order to artificially increase the “failure” rate. As a consequence, we decided to consider any score

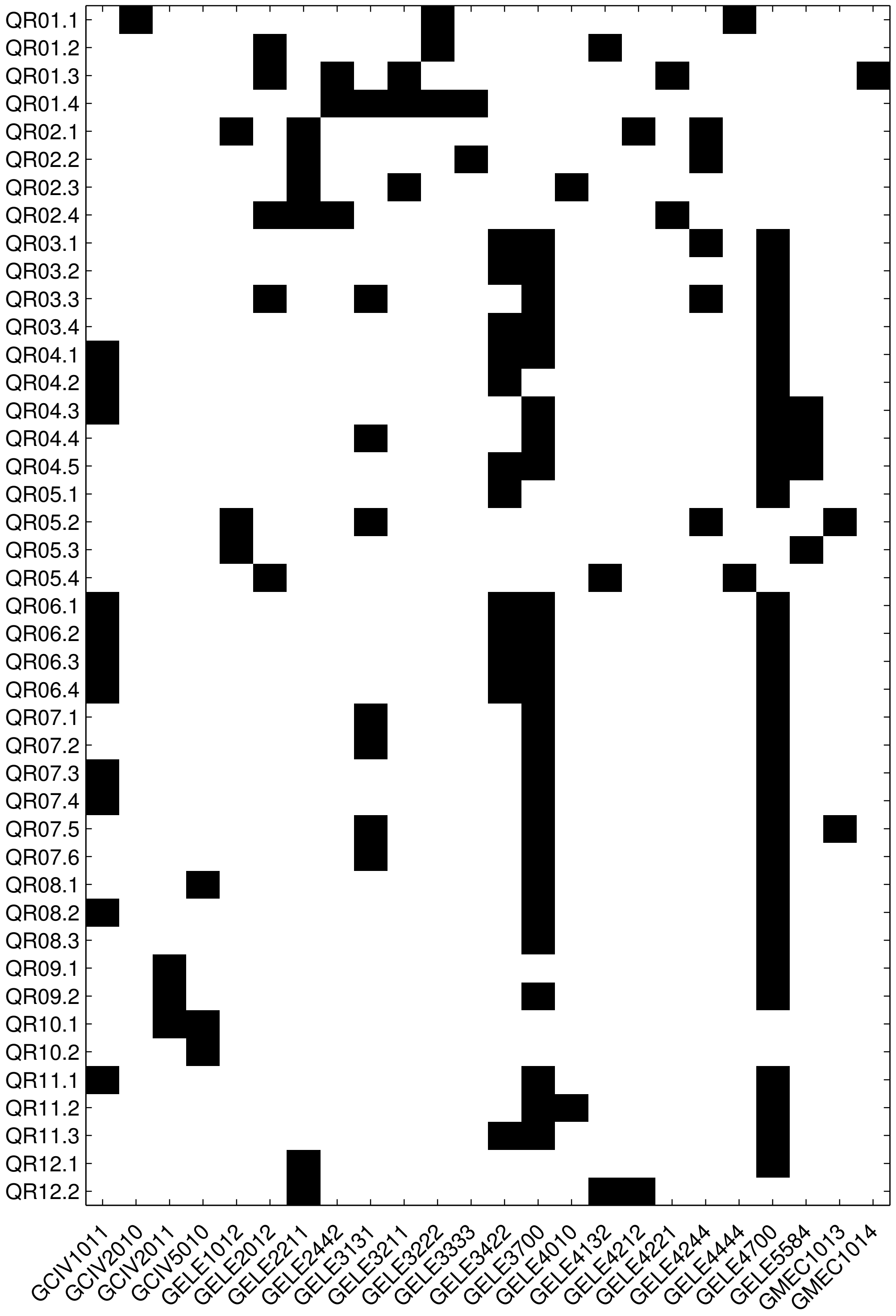

Figure 2: Transpose of the Q-matrix of our use case, mapping 43 skills to 24 items.

below or equal to 2 as a failure. That raised the number of failures to 73 (n = 356 records) for 24 different courses and impacted all students. This change added a larger number of interesting skills to the analysis.

After having defined success and modified consequently our dataset, it was time to calculate opportunity numbers for each student and skills before estimating the AFM parameters. Parameters estimation resulted in student proficiencies α, skills easiness β and learning rates γ required for the analysis.

4.2.3. Analysis of students proficiency

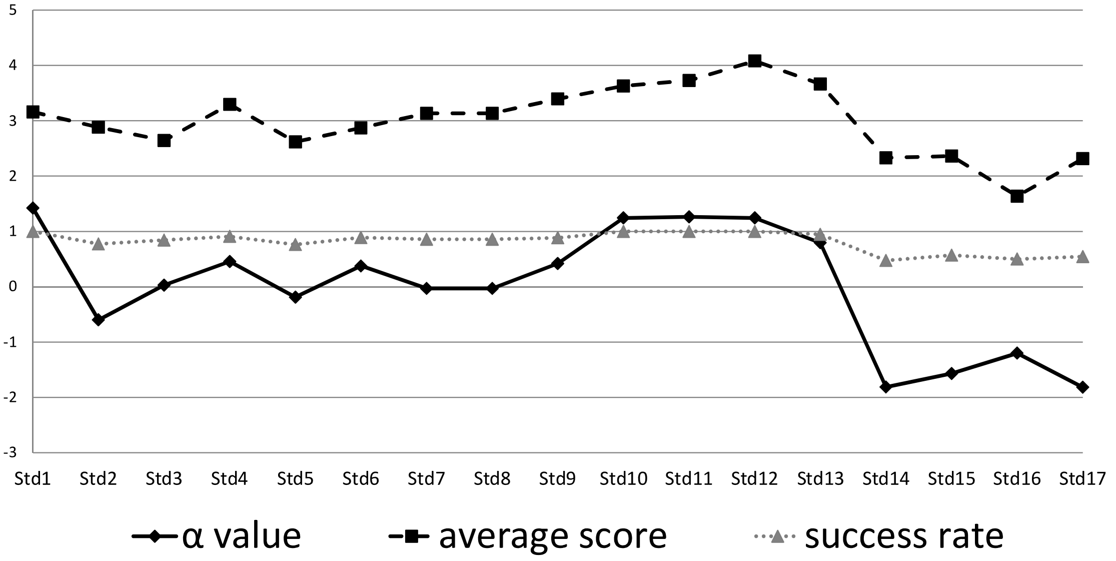

Figure 3: α values calculated with average scores and success rate from first attempts.

Plotting the estimated proficiency α in Figure 3, we can see that the values of α do not necessarily follow the average scores. This may seem counter-intuitive at first sight, but there is usually a good explanation for the observed variations. The first thing to keep in mind is that the average score is a continuous value between 0 and 4.3, while AFM parameters are estimated from binary observations (Success/failure). As a result, if two students succeeded and failed the same courses, they would get the exact same estimated proficiency even though one may have an average of 4.2 while the other an average of 2.1. That being said, success and failure rates on courses can bring other refinements. For instance, the value of α for student Std2 is significantly lower than for Std1 and Std3, while her average score at first attempt is close and between that of Std1 and Std3. The difference among those students is that their success rate is very different. While student Std1 succeeded in all her courses at first attempt, Std2 succeeded only in 77% of her exams and Std3 in 84%. If we consider only the courses they had in common, Std2 success rate drops to 72% while Std3 holds close to 83%. This shows that, on average, while Std2 score has been higher than Std3, her proficiency to learn and master electrical engineering skills may have been slightly lower than Std3. Considering now Std16: we notice that her average score is lower than that of Std15 and Std17 while her α value is higher. Unfortunately, in this situation, the success rate does not completely explain the discrepancy. In this case, Std16 has less success at first attempt than the others (50% compared to 57% for Std15 and 55% for Std17). However, at the end of the session, by passing again failed exams, she ended up with a final success rate (56%) above Std17 (55%) but still slightly below Std16 (57%). In addition, Std16 passed fewer different exams (16 compared to 21 and 23). This student thus had less opportunities to practice skills and may have been exposed to different ones. That would explain Std16's higher proficiency, as estimated by the model. In addition, the model estimate for Std16's α may be less reliable than for other students, because it was obtained from fewer observations. This is an excellent case that illustrates why the interpretation of parameters requires caution, in order to avoid drawing conclusions on entities observed a limited number of times.

4.2.4. Analysis of skills parameters

Estimated skills parameters are presented in Figure 4. A significant number of skills have a β close to zero,like QR08 and QR09, showing a limited learning through opportunities. In addition several skills appears to be very difficult (low β) but easy to learn (high γ), such as QR04.4, QR03.1 and QR11.2 while others, like QR02.4, seem very easy and requires essentially no learning. Finally we can notice that some skills get very similar β and γ values.

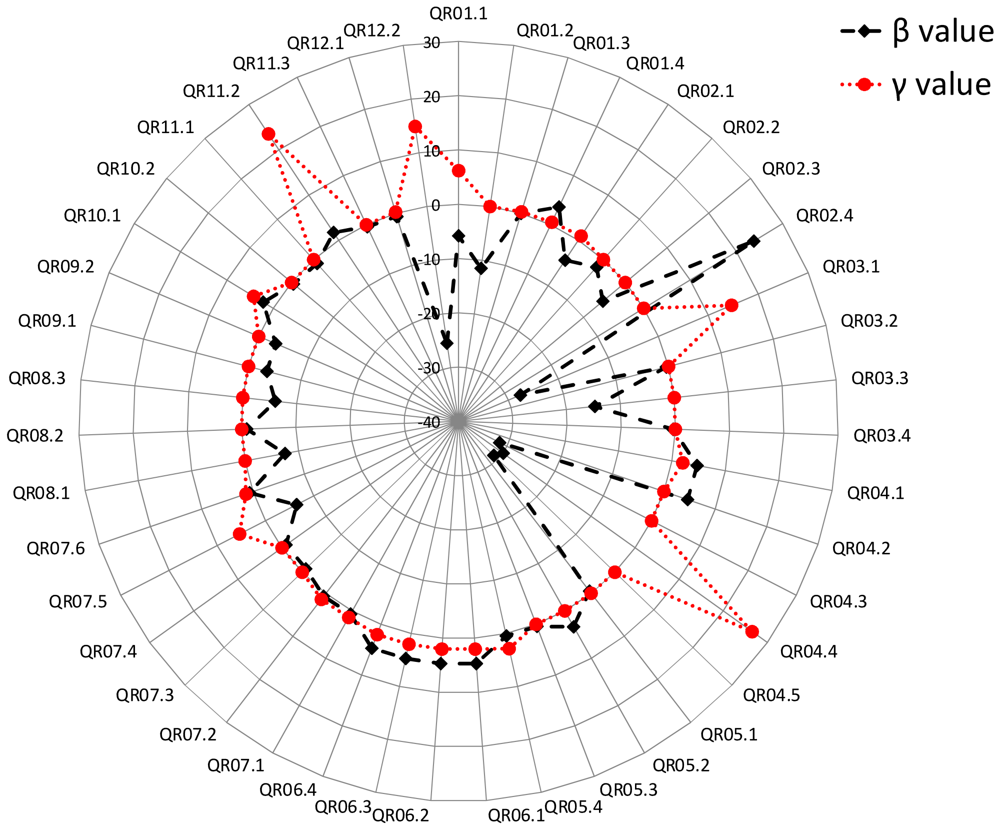

Figure 4: Values of β and γ calculated with Matlab implementation.

This similarity comes from the additive nature of AFM. All knowledge components that are only met together in the same courses get β and γ values that are mostly identical. This is the case, for instance, for all 4 of the skills of attribute QR06 ("Individual and team work") as illustrated in Figure 5. We can see that these skills are always evaluated altogether by the courses GCIV1011, GELE3422, GELE3700 and GELE4700.

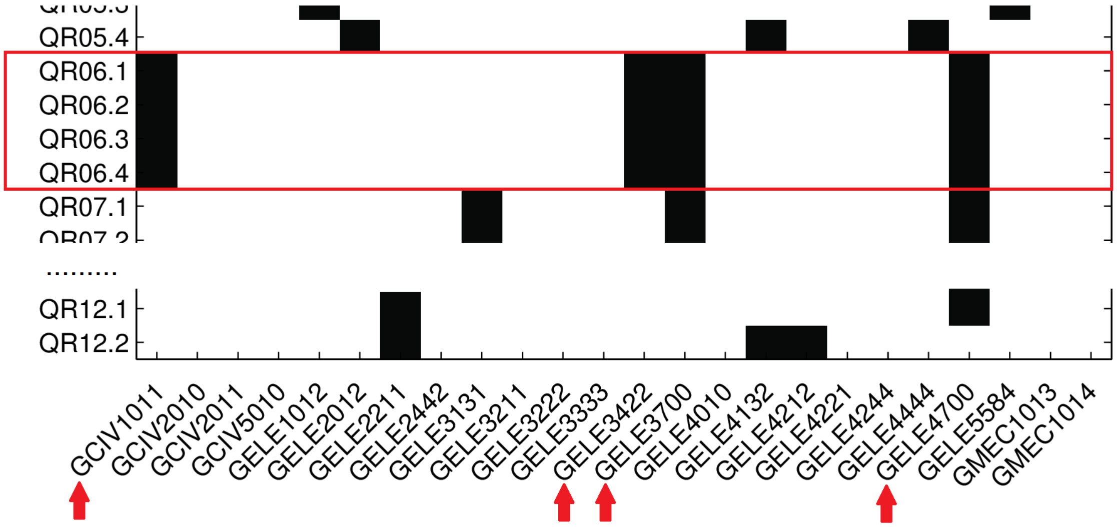

Figure 5: Slice of the Q-matrix (transposed) showing that the four skills for attribute QR06 are always evaluated together in items GCIV1011, GELE3422, GELE3700 and GELE4700.

As they get the same values of β and γ with an error rate close to zero, their learning curves are identical (See Figure 6).

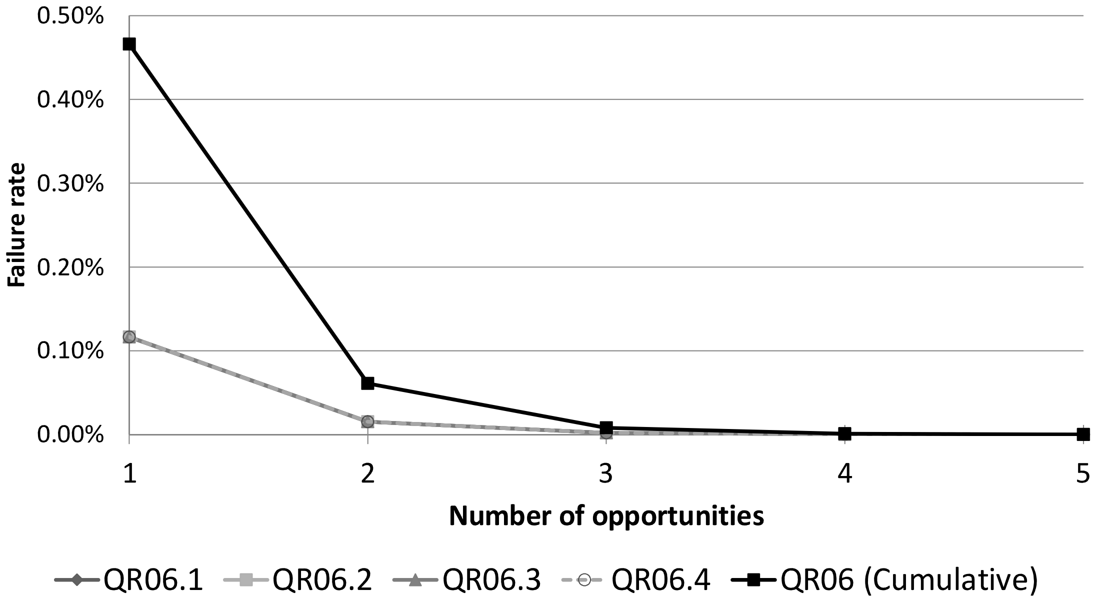

Figure 6: Learning curves of skills from attribute QR06 are superposed as the skills are always evaluated together. It might make sense to modify the Q-matrix and to merge them in one QR06 skill as illustrated by the QR06 cumulative learning curve.

Sometimes the model has also difficulties to finely discriminate skills parameters and tend to compensate by going to the extremes. For instance, the skill QR03.3 is always associated to other skills in different courses and the model tends to attribute potential failures to this skill.

The observed curve in Figure 7 is plotted by computing the average error from all users for each opportunity of skill Qr03.3. As skill QR03.3 is never met alone, the observed curve is compensated by other skills associated to QR03.3, at each opportunity. In cases where a skill is not solely attributed to items, the observed learning curve therefore does not necessarily reflect the characteristics of that one skill. In this case, observed curves reinforce the idea that the difficulty of this skill might have been overstated.

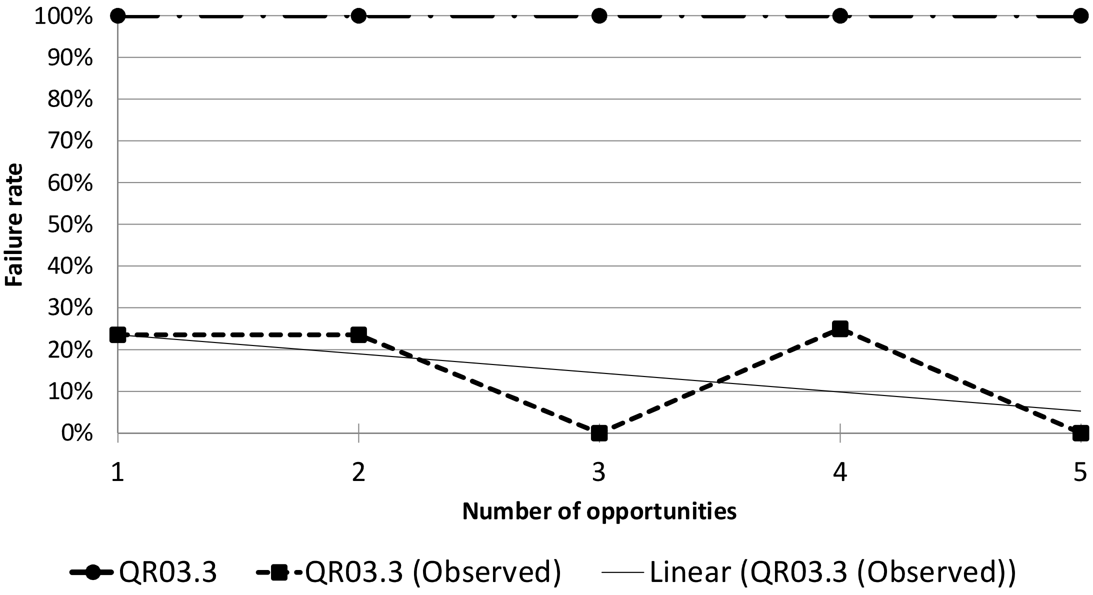

Figure 7: Learning curves of skills QR03.3 from AFM and as observed

Let's now draw more learning curves from the estimated β and γ. while limiting erroneous inferences that can be the result of Q-matrix misconception, data scarcity or simply model limitations. With learning curves, the objective is to understand if the skills envisioned are of the expected difficulty but more importantly if they are mastered at the end of the curriculum. In order to succeed in this exercise, it is necessary to make an appropriate selection. In the examples presented (Fig. 8), we limited our interest to learning curves that meet the following criteria:

- non zero slope (γ) inducing a potential learning;
- realistic easiness with at first opportunity an error rate above 10 percents;
- skills met by most students with an average opportunity number higher than 2.3.

Obviously, the opportunity threshold should vary from one study to the other. For instance, our dataset contained a limited number of observations with a significant number of skills. With more observations, the average opportunity threshold could be chosen higher, ideally above five opportunities. Learning curves for knowledge components meeting those criteria are represented in Figure 8. The plots are limited to five opportunities since no opportunities above this threshold were observed.

Meaningful information can be obtained from the analysis of learning curves in Figure 8. First we can notice that curves for skills QR01.1, QR07.5, QR04.4 and QR03.1 have an error rate close to zero after the first or the second opportunity. It suggests that these skills can be learned quickly for the average user. As a result testing these skills more than three time could prove to be useless.

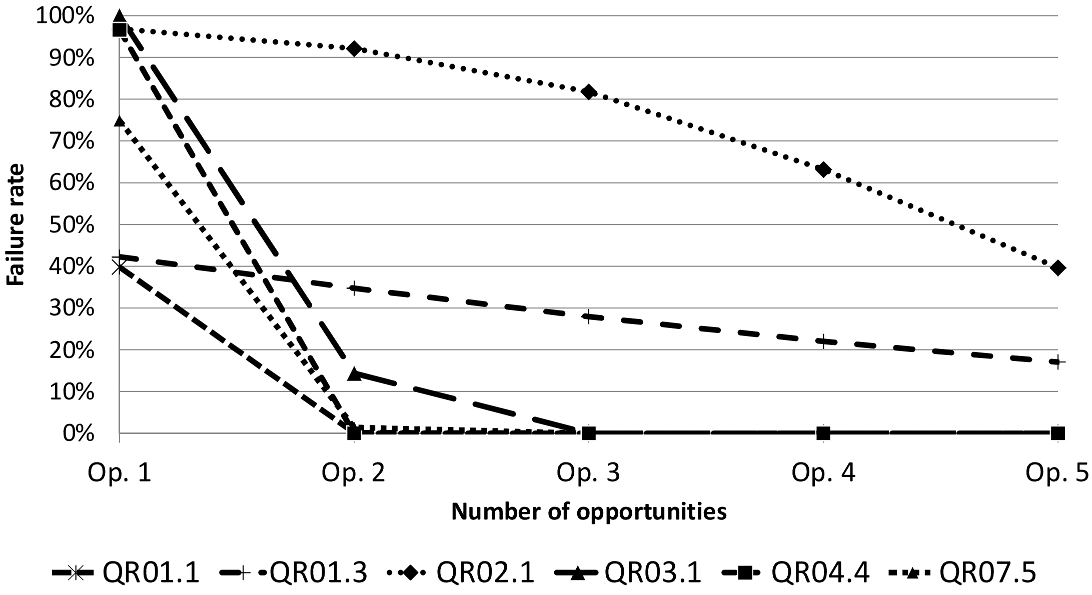

Figure 8: Learning curves of interest

Skills QR02.1 and QR01.3 show interesting patterns. The error rates decrease over time, as expected to provide evidence of learning. However, their shape is far from ideal. The explanation may be coming from the fact that some courses with the same skills have been completed the same semester. For instance, QR01.3 has been met by the same students several times in one semester in courses GELE2442, GELE2012 and GELE3333. However, in order to use AFM, we had to calculate the opportunity number t_{ik}. We therefore had to arbitrarily rank the courses, creating an opportunity sequence were in reality results may have been observed at the same time or in a different order.

4.2.5. Analysis outcomes and limitations

The use case we presented was not originally intended for an analysis with AFM. As a result, it provided us with a limited number of interesting curves. However, the exercise highlighted several flaws of this assessment map that can be immediately addressed to iteratively lead towards evidence based curriculum improvement and certification. Here is the list of our outcomes:

- Limit the number of skills associated to courses. In this assessment maps some courses were associated to up to 25 skills. This is likely to bring easiness (β) parameters to the extreme and amplify a compensation mechanism due to the additive nature of the model. The assessment map for courses GELE3700 and GELE4700 may be particularly worth discussing and revising.
- Each skill should be evaluated separately if possible. Again, because of the compensation mechanism, it is more difficult to find the right parameter value when skills are always

combined. We have seen in our analysis that skills of the attribute QR06 are problematic. If it is not possible to redistribute them in different assessments, an option is to merge them in one skill.

- Increase the observations and or decrease the number of skills in the assessment map. Because of the low opportunity numbers, the majority of the estimated skills parameters were not usable for the analysis. To increase the number of observations, it would make sense to either get more students transcripts or to get more assessments of the current students by looking at the assessments within the courses.

- Limit concurrent courses evaluating the same skills at the same time. The intent here is to improve opportunity number calculation but this may benefit training efficiency as well by limiting redundancy between courses especially for skills of limited difficulty and fast learning.

The outcomes presented above constitute a first step of an iterative process. A natural next step would consist in improving the assessment map structure. Running an analysis with a new version of the Q-matrix may confirm the results obtained with the first version. For instance, we discovered that some skills were very quickly learned and required no further assessments in the curriculum. It may make sense to remove them from several courses, as explained in previous section. Further iterations of this analysis and modifications of the Q-matrix and the courses will bring new valuable information. In this improvement process, it would also make sense to compare models goodness of fit between iterations to get a sense of the progress made. That way AFM can be a precious tool towards an educational “Kaizen” approach supporting a continuous improvement of learning and teaching practices.

Among the major limitations of the presented use case, we can highlight the absence of assessment internal consistency check and of standard errors computation on the model parameters. Not checking internal consistency can be a problem considering that stability of model estimates are known to be particularly weak with a few number of observations as we have in the presented case. In this situation, a very few deviant measures could drastically impact the internal consistency of the assessment instrument. It would be valuable for the AFM end-user to calculate the internal consistency (using for instance Cronbach’s-α (Cronbach, 1951)) and adjust the Q-matrix and associated observations in accordance before applying the AFM model. The second limitation is the absence of standard errors on the model parameters and more particularly on the skills parameters β and γ. Standard errors could provide meaningful information regarding the validity of the inferences made on skills.

5. Conclusion

In this paper, we first situated the Additive Factor Model in the history of Item Response Theory. We introduced it as a multidimensional IRT (Section 2.) that differentiates itself from other models by proposing very unique set of parameters, particularly well suited to skills assessment analysis. We formally presented the model in details, in Section 3.. We explained how to estimate its parameters using off-the-shelf unconstrained optimization on the likelihood function and shared our understanding of the model and its relation to the derivatives of the likelihood function. We also explained how to draw learning curves in order to visually read and understand skills parameters and provided a discussion of constraints on the parameter values, which we hope will be useful to the reader. We believe that providing many technical details is important to get a clear understanding of model's capacities and limitations. We also deliberately provided details on the implementation of AFM parameter estimation, so that it can be easily implemented in Matlab or other high-level languages. Finally, we presented a use case showing how AFM can be used to model and analyze a post-secondary curriculum.

We hope that the level of details and explanation in this article will allow researchers, developers and software engineers to quickly build new tools and produce new research using AFM. The proposed implementation can be done quickly and easily extended to many other models including the Rasch model and others mentioned in the paper. The use case we present also contributes to our intent to illustrate what can be done easily with AFM in a very mainstream context, a post-secondary curriculum as they exist almost everywhere. While the data set we used was clearly not intended for such study, it was still possible to provide meaningful insights on ways to improve the curriculum and the learning experience.

We stressed that using AFM, as any analytical model (Kop et al., 2017), requires practitioners to exercise critical thinking, when looking at the parameters values. More particularly, it would be important to more explicitly characterize parameters uncertainty by computing standard errors (Philipp et al., 2017), something we did not do in our implementation example and that is sometimes missing in AFM related literature. As illustrated, for example, in Section 4.2., the model could be useful to infer meaningful assumption for some knowledge components but it is not the case for all. This suggests issues with either the Q-matrix structure, or the number of observations available for the corresponding items. Although it is tedious to check that each skill fills the appropriate criteria before modeling and inferring, most of these background verification tasks could easily be automated. This would greatly increase AFM usability and broaden its audience while creating a new generation of pedagogical tools.

6. ACKNOWLEDGMENT

This work is part of the National Research Council Canada program Learning and Performance Support Systems (LPSS), which addresses training, development and performance support in all industry sectors, including education, oil and gas, policing, military and medical devices.

REFERENCES

BARNES, T. 2005. The Q-matrix method: Mining student response data for knowledge. In *AAAI Educational Data Mining workshop*. Pittsburgh, PA, 39.

BIRENBAUM, M., KELLY, A. E., AND TATSUOKA, K. K. 1992. *Diagnosing Knowledge States in Algebra Using the Rule Space Model*. Educational Testing Service Princeton, NJ: ETS research report. Educational Testing Service.

BIRNBAUM, A. 1968. Some latent trait models and their use in inferring an examinee s ability. In *Statistical Theories of Mental Test Scores*, F. Lord and M. Novick, Eds.

BOCK, R. D. 1997. A brief history of item theory response. *Educational Measurement: Issues and Practice 16,* 4, 21–33.

CANADIAN ENGINEERING ACCREDITATION BOARD. 2014. Accreditation criteria and procedures. `http://www.engineerscanada.ca/sites/default/files/2014_accreditation_criteria_and_procedures_v06.pdf` [last accessed February 2017].

CEN, H. 2009. Phd thesis. Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

CEN, H., KOEDINGER, K., AND JUNKER, B. 2006. Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems: 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006. Proceedings*, M. Ikeda, K. D. Ashley, and T.-W. Chan, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 164–175.

CEN, H., KOEDINGER, K., AND JUNKER, B. 2007. Is overpractice necessary? — improving learning efficiency with the cognitive tutor through educational data mining. In *Proceeding of the 2007 Conference on Artificial intelligence in Education: Building Technology Rich Learning Contexts that Work*, R. Luckin, K. R. Koedinger, and J. Greer, Eds. Number 158 in Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, Netherlands, 511–518.

CEN, H., KOEDINGER, K., AND JUNKER, B. 2008. Comparing two irt models for conjunctive skills. In *Proceedings of the 9th international Conference on intelligent Tutoring Systems*, B. P. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, Eds. Lecture Notes In Computer Science. Springer-Verlag, Berlin, Heidelberg, 796–798.

CORBETT, A. AND ANDERSON, J. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction 4,* 4, 253–278.

CRONBACH, L. J. 1951. Coefficient alpha and the internal structure of tests. *Psychometrika 16,* 3 (Sep), 297–334.

DE LA TORRE, J. 2008. An empirically based method of q-matrix validation for the dina model: Development and applications. *Journal of Educational Measurement 45,* 4, 343–362.

DE LA TORRE, J. 2009. DINA model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics 34,* 1 (Mar.), 115–130.

DESMARAIS, M. 2011a. Conditions for effectively deriving a Q-matrix from data with non-negative matrix factorization. In *4th International Conference on Edu- cational Data Mining*, C. Conati, S. Ventura, T. Calders, and M. Pechenizkiy, Eds. Eindhoven, Netherlands, 41–50.

DESMARAIS, M. 2011b. Mapping questions items to skills with non-negative matrix factorization. *ACM-KDD-Explorations 13,* 2, 30–36.

DESMARAIS, M. AND NACEUR, R. 2013. A matrix factorization method for mapping items to skills and for enhancing expert-based Q-matrices. In *Artificial Intelligence in Education*, H. Lane, K. Yacef, J. Mostow, and P. Pavlik, Eds. Lecture Notes in Computer Science, vol. 7926. Springer Berlin Heidelberg, 441–450.

DURAND, G., BELACEL, N., AND GOUTTE, C. 2015. Evaluation of expert-based q-matrices predictive quality in matrix factorization models. In *Design for Teaching and Learning in a Networked World, EC-TEL 2015 conference*. Springer, 56–69.

FLETCHER, R. 1987. *Practical Methods of Optimization*. Wiley.

JUNKER, B. W. AND SIJTSMA, K. 2001. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement 25,* 3, 258–272.

KOEDINGER, K., BAKER, R., CUNNINGHAM, K., SKOGSHOLM, A., LEBER, B., AND STAMPER, J. 2010. A data repository for the EDM community: The PSLC Datashop. In *Handbook of Educational Data Mining*, C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker, Eds. CRC Press.

KOP, R., FOURNIER, H., AND DURAND, G. 2017. A Critical Perspective on Learning Analytics and Educational Data Mining. In *The Handbook of Learning Analytics*, C. Lang, G. Siemens, A. F. Wise, and D. Gaševic, Eds. Society for Learning Analytics Research (SoLAR), 319–326.

LI, Y., JIAO, H., AND LISSITZ, R. 2012. Applying multidimensional item response theory models in validating test dimensionality: An example of k–12 large-scale science assessment. *Association of Test Publishers 13,* 2.

LIU, J., XU, G., AND YING, Z. 2012. Data-driven learning of Q-matrix. *Applied Psychological Measurement 36,* 7, 548–564.

LORD, F. 1952. A theory of test scores. *Psychometric Monograph 7.*

MURRE, J. M. AND CHESSA, A. 2011. Power laws from individual differences in learning and forgetting: mathematical analyses. *Psychonomic bulletin & review 18,* 3, 592–597.

NEWELL, A. AND ROSENBLOOM, P. S. 1981. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition 1*, 1–55.

PHILIPP, M., STROBL, C., DE LA TORRE, J., AND ZEILEIS, A. 2017. On the estimation of standard errors in cognitive diagnosis models. *Journal of Educational and Behavioral Statistics*.

RASMUSSEN, C. E. 2006. `minimize.m`. `http://learning.eng.cam.ac.uk/carl/code/minimize/minimize.m`, [last accessed February 2017].

RECKASE, M. D. 1997. A linear logistic multidimensional model for dichotomous item response data. In *Handbook of Modern Item Response Theory*, W. van der Linden and R. Hambleton, Eds. Springer New York, 271–286.

RITTER, F. AND SCHOOLER, L. 2001. Learning Curve, The. In *International Encyclopedia of the Social & Behavioral Sciences*, Editors-in-Chief: Neil J. Smelser and Paul B. Baltes, Eds. Pergamon, Oxford, 8602–8605.

RIVERS, K., HARPSTEAD, E., AND KOEDINGER, K. 2016. Learning curve analysis for programming: Which concepts do students struggle with? In *Proceedings of the 2016 ACM Conference on International Computing Education Research*. ICER '16. ACM, New York, NY, USA, 143–151.

RUPP, A. A. AND TEMPLIN, J. 2008. The effects of q-matrix misspecification on parameter estimates and classification accuracy in the dina model. *Educational and Psychological Measurement 68,* 1, 78–96.

SIJTSMA, K. AND JUNKER, B. W. 2006. Item response theory : Past performance, present developments, and future expectations. *Behaviormetrika 33,* 1 (jan), 75–102.

STAMPER, J. AND KOEDINGER, K. 2011. Human-machine student model discovery and improvement using DataShop. In *Artificial Intelligence in Education*. Springer Berlin Heidelberg, 353–360.

SUN, Y., YE, S., INOUE, S., AND SUN, Y. 2014. Alternating recursive method for Q-matrix learning. In *7th Intl. Conf. on Educational Data Mining*. 14–20.

TATSUOKA, K. 1983. Rule space: an approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement 20,* 4, 345–354.

TATSUOKA, K. K. 1984. Analysis of errors in fraction addition and subtraction problems. Final report, Computer-based Education Research Laboratory, University of Illinois at Urbana-Champaign.

THURSTONE, L. L. 1925. A method of scaling psychological and educational tests. *Journal of Educational Psychology 16,* 1, 433–451.