Exploration of the Robustness and Generalizability of the Additive Factors Model

Tomáš Effenberger
tomas.effenberger@mail.muni.cz
Masaryk University
Brno, Czech Republic

Radek Pelánek
xpelanek@fi.muni.cz
Masaryk University
Brno, Czech Republic

Jaroslav Čechák
xcechak1@fi.muni.cz
Masaryk University
Brno, Czech Republic

ABSTRACT

Additive Factors Model is a widely used student model, which is primarily used for refining knowledge component models (Q-matrices). We explore the robustness and generalizability of the model. We explicitly formulate simplifying assumptions that the model makes and we discuss methods for visualizing learning curves based on the model. We also report on an application of the model to data from a learning system for introductory programming; these experiments illustrate possibly misleading interpretation of model results due to differences in item difficulty. Overall, our results show that greater care has to be taken in the application of the model and in the interpretation of results obtained with the model.

CCS CONCEPTS

• **Human-centered computing** → **User models**; • **Applied computing** → **Interactive learning environments**.

KEYWORDS

student modeling; learning curves; knowledge components; introductory programming

ACM Reference Format:
Tomáš Effenberger, Radek Pelánek, and Jaroslav Čechák. 2020. Exploration of the Robustness and Generalizability of the Additive Factors Model. In *Proceedings of the 10th International Conference on Learning Analytics and Knowledge (LAK '20), March 23–27, 2020, Frankfurt, Germany*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3375462.3375491

1 INTRODUCTION

Student modeling is a key element of the development of intelligent learning systems. Currently, there are two main families of student models: hidden Markov models (including the much-studied Bayesian Knowledge Tracing model) and logistic models [26]. One widely used model from the logistic family is the Additive Factors Model (AFM). A key feature of AFM is that it considers several knowledge components (skills) for each item.

AFM is mostly used in a different way than other models. A typical application of a student model is to provide estimates of student skills and the probability that the next answer will be correct [26]. These estimates are used to guide the online behavior of an adaptive system, e.g., to choose questions of suitable difficulty or to serve as a mastery criterion. AFM is used for a different purpose—an offline analysis of data with the goal to refine the domain model, specifically the mapping of items to knowledge components, which is often called Q-matrix. Table 1 provides a simple example of such a Q-matrix. The example also illustrates typical questions that we face when specifying a Q-matrix: Should + and − be two separate knowledge components, or is it better to merge them? Should we introduce additional knowledge components, e.g., "priorities of operators" or "working with negative numbers"? In a typical application of AFM, the model is used for answering such questions by analysis of data about student performance.

Table 1: An example of a Q-matrix.

	−	+	×	()
$10 + 3 \times 2$	0	1	1	0
$(7 - 4) \times 3$	1	0	1	1
$2 + (3 + 5)$	0	1	0	1
$8 - (6 + 2)$	1	1	0	1
$5 - 2 \times 6$	1	0	1	0

AFM has been used in many studies, e.g., [1, 16, 17, 19–21, 23, 29] (we provide a more detailed discussion of these studies below). Most of these studies do not pay much attention to methodological details of parameter fitting and model comparison, e.g., the specific division of data into training and testing set, or the choice of metrics. Recent research, however, shows that such details may be important [27]. AFM also makes many simplifying assumptions about the learning process. These assumptions are not taken into account or even explicitly acknowledged in many AFM studies. It is thus not clear, how robust are the results reported in existing studies and whether the reported results generalize to other settings.

The purpose of this paper is thus to explore the robustness and generalizability of the Additive Factors Model. We aim

LAK '20, March 23–27, 2020, Frankfurt, Germany

ACM ISBN 978-1-4503-7712-6/20/03…$15.00
https://doi.org/10.1145/3375462.3375491

to identify and clearly describe the limitations of the model and potentially dangerous consequences of ignoring these limitations (e.g., misleading interpretation of results). We also focus on learning curves, which are often used to visualize the results obtained using AFM, but their exact computation is typically not specified. We discuss four types of learning curves and their (dis)advantages.

To explore the model, we use data from a learning system for introductory programming. This dataset has several aspects that are quite common and were not much studied in previous research with AFM, e.g., an important role of response times or the presence of attrition bias in data. These characteristics enable us to check the generalizability of the model.

Our explorations show that the results obtained with the AFM have to be interpreted carefully. Previous research often takes several different Q-matrices, fits AFM parameters for these Q-matrices, compares the predictive accuracy of the fitted models, and concludes that the Q-matrix with the best fit provides the best model of the studied domain. This can be misleading when the underlying assumptions of AFM are not satisfied. Specifically, the model ignores differences in the difficulty of items that belong to the same knowledge component. If the data contain such differences, the model may use knowledge components in Q-matrix just to compensate for the missing difficulty parameter. Our results show that even the addition of a purely random knowledge component can sometimes improve model predictions.

2 THE BASIC ADDITIVE FACTORS MODEL

The Additive Factors Model has been proposed by Cen et al. [3] provides a discussion of the basic version of AFM and proposes moreover "learning factors analysis" (LFA), which combines AFM with an automated A* search to find the best model. [2] provides an extended discussion of AFM and LFA. A recent review [5] provides a good summary of the model and description of details of model parameter fitting (which are missing in most other papers using AFM). We follow the notation used in this review (which is mostly the same as in other AFM papers).

2.1 Model Formulation

Additive Factors Model predicts the probability that a student i will answer an item j correctly. The model is multidimensional—each item can be mapped to several knowledge components (through a Q-matrix), parameters of these knowledge components influence the predicted probability of success. The model is described by the following equation:

$$P(Y_{ij}|\alpha,\beta,\gamma) = \sigma\left(\alpha_i + \sum_{k=1}^{K}\beta_k q_{jk} + \sum_{k=1}^{K}\gamma_k q_{jk} t_{ik}\right)$$

where:

- i is student index, j is item index,
- Y_{ij} is the binary response of a student i on a item j,
- $\sigma(x) = 1/(1+e^{-x})$ is the standard logistic function,
- K is the number of skills, J is the number of items,
- Q is the $J \times K$ binary matrix – q_{jk} is the indicator that item j uses skill k,
- α_i is the proficiency (prior skill) of a student i,
- β_k is the easiness of skill k,
- γ_k is the learning rate for skill k,
- t_{ik} is the number of times student i has practiced skill k (opportunity count).

The model can be seen as an extension of basic Item response theory models (specifically Rasch model) with learning (for the discussion of relation to IRT see [5]).

2.2 Model Assumptions

AFM makes many simplifying assumptions, which are clearly not satisfied in practical situations. This is not necessarily a problem since all models are wrong and the use of simplification in modeling learning is unavoidable. It is, however, useful to explicitly formulate the used simplifying assumptions and to be aware of them when using the model.

The main simplifying assumptions are the following:

- Students differ only in a unidimensional prior skill. The differences in prior skill are not specific to individual knowledge components. All students have the same learning rates.
- All practice opportunities are considered equal. Student performance is not taken into account.
- Learning is linear (on the logit scale).
- Observed outcomes are binary. Other aspects of student performance, like "response time" or "near miss answers", are either not available or are ignored.
- The relation of items to skills is binary, i.e., we cannot express the fact that some item is related to one skill more significantly than to others.
- The model uses a compensatory model of skills, i.e., insufficient practice of one skill can be compensated by the practice of another skill.
- The model ignores the difficulty of items, i.e., two items with the same mapping to skills are considered interchangeable.
- The model does not take into account any biases in data, e.g., a mastery attrition bias (which occurs when the number of items solved is correlated with prior skill).

There exist extensions of the model which address some of these limitations (see discussion in Section 3.2), but most studies that applied the model use the basic version.

2.3 Parameter Fitting

To use the model, we need to estimate its parameters. Specifically, we need to estimate the vectors α (student skills), β (easiness of skills) and γ (learning rate for skills) based on the available data on student performance (Y_{ij}, t_{ij}) and the provided matrix Q. The parameter estimation can be done by maximizing the penalized likelihood (with a regularization term on α), see [5] for detailed derivation.

There are off-the-shelf tools for maximizing the penalized likelihood, but care has to be taken to prepare the data for the optimizer properly. For example, if practice opportunities counts are disproportionately high (compared to the other features in the model, which are just binary), it might be crucial to normalize them to the 0–1 range, in order for the optimizer to pay attention to all parameters. We discuss other methodology choices in Section 5.3.

The basic applications of AFM are to fit a model and then explore the obtained learning curves (based on β, γ values) and to compare models based on different Q matrices to see which one is better, i.e., which Q matrix better describes the learning domain.

2.4 Model Comparison

To compare models, we need to quantify their predictive ability. Previous research used two approaches for such comparison. The first approach is to use metrics like the Akaike information criterion (AIC) and Bayesian information criterion (BIC), which combine the likelihood of observations with a penalty for the number of parameters. The second approach is to use cross-validation, where the model is trained on a subset of the data and evaluated on a separate test set. The cross-validation approach is a better estimation of the ability of the model to generalize but is more computationally demanding, which may be a problem in some applications of AFM (specifically the Learning Factors Analysis, which performs a search over Q-matrices).

The cross-validation approach involves several decisions, which are often not reported in previous studies using AFM. We have to divide data into a training set and testing set. This can be done in several ways [27]. In the case of AFM applications, a natural approach is to use student-level division, i.e., all observations for a single student are included either in a training set, or a test set. With this kind of data division, it is not clear how to treat the α parameters. Since the training and testing set do not share student, the trained α parameters are not directly useful for predictions on the testing set (their role is to enable us to get better estimates of β and γ parameters). For students in the testing set, we do not have any α estimates. Assuming that α parameters are regularized in parameter fitting, we can use 0 as a reasonable default value for α for students in the testing set. It would be possible to update the α estimate after each observation in the testing set by finding a maximum likelihood estimate. Such update, however, is not trivial and it does not seem that any previous work using AFM uses this approach. Another decision concerns the choice of a performance metric; a typical choice is the Root Mean Square Error (RMSE) metric, although other choices are possible [25].

2.5 Methodological Issues

Most AFM applications rely on parameter fitting support in DataShop [15] and do not consider methodological issues in great detail. However, few recent works have explored the potential impact of biases in data on parameter fitting of AFM.

Käser et al. [14] studied the impact of biases, specifically of the mastery attrition bias. They analyzed the impact of the bias on model fit for AFM and several of its variants. An interesting aspect of this work is that they consider not only the common cross-validation evaluation but also the prediction of student re-test results. Goutte et al. [13] used simulated data to analyze the impact of attrition bias on learning curves produced using AFM; they also propose an alternative way to compute learning curves and test it using both simulated and real data. Durand et al. [6] focus on the issue of parameter fitting in AFM, estimating not just the parameter values, but also standard errors of model parameters.

3 USE OF AFM IN PREVIOUS WORK

After the overview of the basic Additive Factors Model, we now provide the discussion of its applications. We also mention several of its extensions and variations.

3.1 Applications of the Model

The model has been applied quite widely in recent years, mostly in its basic form. Aleven and Koedinger [1] and Liu and Koedinger [19] provide a high-level discussion of knowledge component modeling, including the use of AFM.

Examples of specific analysis are provided by [16], who describe analyzes of 11 datasets which leads to finding better models (with respect to predictive accuracy). To really show that the newly found models are "better", it is necessary to apply them in a learning system and show an improvement in learning gains of students. This type of analysis has been called "closing the loop". One typical early example is by Koedinger et al. [17], who used AFM to identify poorly fitting knowledge components and to propose possible improvement. Based on the model results, the tutoring system was redesign, and the intervention was tested using an experiment; the results showed better learning (as measured by a post-test).

In a similar manner, AFM has been used in other studies, mostly for analysis of data from learning systems for mathematics. Liu et al. [20] used Learning Factors Analysis to find model improvements in the domain of geometry. Long et al. [21] compared several manually specified Q-matrices in the domain of algebra (solving equations); their conclusion was that more fine-grained models are better. Nguyen et al. [23] presented a similar type of analysis for the topic of fractions and decimals. Rivers et al. [29] used AFM to fit data from programming; focusing on analyzing learning curves for different concepts, trying to find out "what students struggle with". All these works are based on support for AFM available in DataShop [15].

3.2 Extensions and Variants

The basic model is based on many simplifying assumptions. The model can be extended to remove some of these simplifications. Several such extensions have been explored in previous work.

AFM takes into account only the number of student's attempts on a knowledge component. Performance Factor Analysis [24] is an extension of the basic model that takes the student performance into account—distinguishing correct and incorrect answers. This approach has been further extended to take recency of attempts into account [11]. From a modeling point of view, PFA is only a slight extension of AFM. Considering the practical usage, the two models are quite different. AFM is used mainly for refinement of domain models (where the predictions for individual students are not of primary interest), whereas PFA is used more in the context of estimation of student skills; in previous research, it has been often compared to Bayesian knowledge tracing [12].

Another slight extension of the model, this time more in the domain modeling directions, is the extension with instructional steps. Instructional Factor Analysis [4] adds a new element to the model—the number of "tell steps", i.e., occasions where the student was provided with some information from the system without actively giving an answer.

Another extension addresses one of the simplifying assumptions of the model: "all students learn at the same rate". Liu and Koedinger [18] studied an extension of AFM with individualized student learning rates. The extended model, however, did not lead to improved results; the extension introduces a large number of new parameters which are hard to fit. They also tried clustering of students into groups and introducing group learning rates; this variant led to an improvement.

4 LEARNING CURVES

Once fitted, AFM is often visualized and interpreted using learning curves, which display how the failure probability for individual concepts depends on the practice opportunity count (e.g., [16, 21, 23, 29]). There are several reasonable ways to compute learning curves, each having some considerations and disadvantages. In previous research, a description of the computation is often not reported, although, as we show in this section, different computations can lead to significantly different learning curves. The different types of learning curves also require different interpretation, so stating the type of the learning curve is important to avoid wrong conclusions.

Four basic types of learning curves are summarized in Table 2. All these curves estimate the probability of failure for a given concept k and practice opportunity count t; the difference is in the data used for the estimation. The definitions of these learning curves in the following text are adapted from [13], but are made more explicit to avoid possible ambiguity. *Attempt* refers to the information about the item and student history needed by a fitted AFM for prediction: index of the student (s_a), which concepts are contained in the item ($\vec{q}_a$), and how many times the student practiced each of them before ($\vec{t}_a$). *Opportunity* t denotes specifically the number of previous practice opportunities on items containing the considered concept k. This information is already a part of the attempt but is sometimes treated differently from the rest.

Table 2: The types of learning curves according to which data for its computation are observed, simulated, or predicted by the model. Column *Opportunity* denotes the number of practice opportunities for the considered concept.

Type	Attempt	Opportunity	Success
empirical	observed	observed	observed
marginal	observed	observed	predicted
completed	observed	simulated	predicted
idealized	simulated	simulated	predicted

We use $P(s_a, \vec{q}_a, \vec{t}_a)$ to denote the predicted probability of success for an attempt a of a student s_a to solve an item containing concepts encoded by vector $\vec{q}_a$ if the student had had (at the given moment) the numbers of previous practice opportunities according to the vector $\vec{t}_a$:

$$P(s_a, \vec{q}_a, \vec{t}_a) = \sigma(\alpha_{s_a} + \vec{\beta} \cdot \vec{q}_a + \vec{\gamma} \cdot (\vec{q}_a \odot \vec{t}_a))$$

where α, β, γ are the fitted parameters of the examined AFM and $\odot$ is element-wise multiplication.

To illustrate the definitions of various learning curves and their differences, we use a simulated experiment. In the first step, we generate attempts by an AFM for 300 students and 15 items. The first 5 items contain a single concept k_1 with $\beta = 2, \gamma = 0.1$ (easy from the beginning, but only small improvements after repeated exposures to the concept). The next 10 items contain k_1 and another, more difficult concept k_2 with $\beta = -2, \gamma = 0.3$. All students have the same prior knowledge $\alpha_i = 0$. In the second step, we use the same AFM as was used for data generation (i.e., the true model) to make predictions about the failure of each attempt.

Figure 1 shows the four different learning curves for the concept k_1 in two scenarios (the first 10 practice opportunities). In the first scenario (on the left), items are visited in random order; in the second (on the right), the items are visited in the fixed order, with 10% chance to skip each item. We discuss the properties of individual learning curves in the following subsections; for now, note how much differently can different types of learning curve appear and how their shape can be affected by data collection biases. Specifically, the second scenario illustrates how the item ordering bias can cause some learning curves to increase, even if the predictions are made by the true model.

4.1 Empirical learning curve

Empirical learning curve uses only the observed performances and computes their average for all attempts in which the item contained the given concept and the student had had

Figure 1: Learning curves for data generated by an AFM and predictions made by the same model. The right scenario illustrates effect of item ordering bias.

previously exactly the given number of opportunities to practice this concept. We denote this set of all attempts collected for the given concept k at practice opportunity count t as A_{kt} and the observed binary success of attempt a as o_a:

$$LC_k^E(t) = \text{avg}(1 - o_a \mid a \in A_{kt})$$

The empirical curve is a direct display of the collected data, without the intermediate step of fitting a model. As such, it has the most straightforward interpretation, but it is also the most susceptible to biases and noise in the data.

4.2 Marginal learning curve

Marginal learning curve uses the predicted performances of the fitted AFM; otherwise, the formula is the same as for the empirical learning curve.

$$LC_k^M(t) = \text{avg}(1 - P(s_a, \vec{q}_a, \vec{t}_a) \mid a \in A_{kt})$$

This is sometimes called a *model curve* in previous AFM studies [13] and *marginal plot* or *M-plot* in other contexts [22]. It is usually used to visually assess how well the model fits the observed data. As it is basically just a smoothed version of the empirical curve, it is similarly susceptible to all data collection biases, even if the model itself is unbiased. For example, even if the explored concept does not influence the failure rate at all, the marginal learning curve would not be constantly zero because of other concepts whose occurrence correlates with this one. Figure 1 illustrates how a correlated concept can inflate a visualized difficulty (left) and how the item ordering bias can lead to an increasing marginal learning curve (right).

4.3 Completed learning curve

Completed learning curve uses the collected attempts on the concept k ($A_k = \bigcup_t A_{kt}$), but ignores the observed number of previous opportunities, substituting $0, 1, 2, \ldots, N$, where N is the highest practice opportunity count to display in the learning curve plot. This process can be viewed as data augmentation, replicating each observed attempt $(N + 1)$ times and changing its value of t_k, followed by computing the average prediction as for the marginal learning curve. We use notation $P(s_a, \vec{q}_a, \vec{t}_a, t_k{=}t)$ to denote the prediction of the AFM for the attempt a in which the practice opportunity for the concept k is replaced by value t:

$$LC_k^C(t) = \text{avg}(1 - P(s_a, \vec{q}_a, \vec{t}_a, t_k{=}t) \mid a \in A_k)$$

To understand the difference between the marginal and completed learning curves, consider a Q-matrix with only two concepts, k_1 and k_2 and assume we want to compute the failure rate for concept k_1 at the practice opportunity $t = 3$. The marginal learning curve would estimate the failure rate from attempts with $k_1 = 1$ and $t_1 = 3$ (and arbitrary values of k_2 and t_2), while the completed learning curve would estimate the failure rate using all attempts with $k_1 = 1$ (and arbitrary values of t_1, k_2 and t_2), substituting $t_1 = 3$ before computing the predictions.

This definition differs from the definition of the completed learning curve in [13], in which attempts are simulated only after the last observed attempt of each student, and observed success is used rather than the predicted one whenever available. However, it is not clear (and not specified in the paper), how to set $\vec{q}_a, \vec{t}_a$ for the imputed attempts.

The completed learning curve (as defined in our paper) can be understood as an instance of a more general *partial dependence plot* [10], which is used in the area of interpretability of machine learning models, here applied to slice of data for a given concept k.

Advantage of the completed learning curve is that it uses the same population of items for each practice opportunity count, diminishing the effect of item ordering and attrition biases on the visualization, as can be seen in Figure 1 (right). Of course, it only helps if the model itself is not biased. Another caveat is that in the process of data augmentation, it creates possibly unrealistic attempts with unlikely combinations of practice opportunities for concepts whose occurrence correlates.

4.4 Idealized learning curve

Idealized learning curve does not use any observed data at all. It computes the probability of failure for an imaginary item that only contains a single concept and an imaginary average student s_0 with $\alpha_{s_0} = 0$:

$$LC_k^I(t) = 1 - P(s_0, \vec{q}{=}\vec{0}, q_k{=}1, \vec{t}{=}\vec{0}, t_k{=}t) = \sigma(\beta_k + \gamma_k t)$$

Not using any observed data means no influence of data collection biases, but, again, only provided that the model itself is not biased. While the interpretation of this learning curve is simple, it might not be much useful, because the imaginary item is unrealistic, as the actual items usually contain multiple concepts.

5 APPLICATION TO INTRODUCTORY PROGRAMMING DATA

In this section, we apply AFM to real data from introductory programming. As in previous studies, we define several Q-matrices, fit them to a subset of our data, and evaluate

Table 3: Evaluated Q-matrices and covered concepts. Game concepts are emphasized by *italics*.

Name	Concepts
Q1	*teleports*, *collectables*, *obstacles*, *destructibles*, *program length limit*, sequences, while, repeat, loop, nested-loops, if, else, test, nested control structures, comparison
Q2	*teleports*, *collectables*, *obstacles*, *destructibles*, sequences, while, repeat, nested loops, if, else
Q3	sequences, while, repeat, nested loops, if, else
Q4	sequences, loop, nested loops, test

the fitted models on the remaining test set to see which model performs best. The best AFM is then used to identify relevant concepts, their difficulties, and learning curves. Our goal in this study, however, is not to obtain conclusions about a suitable domain model for learning introductory programming; rather, we are interested in how well AFM generalizes to this new context. Therefore, we repeat this experiment with various methodology choices to see their influence on the potential interpretations of the results.

5.1 Data

We use data from RoboMission (`en.robomise.cz`), an adaptive system for learning introductory programming [8]. In RoboMission, students build programs using a block-based programming interface to solve problems in a grid microworld. The items practice basic programming concepts like sequences of commands, conditional statements, and loops. The system uses adaptive recommendations of items to solve. The data contains 85,000 attempts of 5,800 students to 85 items [7]. In these analyses, we use only the correctness information about each attempt, i.e., whether the student eventually solved the item or not.

5.2 Models

We evaluate three variants of AFM: (1) AFM-BG, which is a simplified AFM without student prior skills (i.e., $\alpha_i = 0$ for all i), (2) AFM-ABG, which is a full AFM, and (3) AFM-BGT, which is AFM-BG with an additional parameter per each item, allowing for different difficulties of items with the same concepts; the motivation for this extension is explained in Section 5.4. Each of the three AFM variants is used with four Q-matrices of different granularity. Table 3 lists the concepts covered by each Q-matrix.

Additionally, we compare the performance of AFMs to two baseline models. The first is *global-avg*, which always predicts the global success rate (computed from all attempts in the training set), and the second is *item-avg*, which predicts the success rate per item.

5.3 Methodology Choices

There is a myriad of methodology choices that one needs to decide in order to fit and compare multiple AFMs. We are interested in how critical these choices are, i.e., which choices could influence the ordering of the fitted models with different Q-matrices, and consequently the interpretation about the domain model. Broadly, these choices can be divided into three categories: data preprocessing, fitting, and evaluation.

Data preprocessing includes filtering, transformations, and resampling. Repeated attempts are usually filtered, as well as students or items with too few attempts. We have found necessary to normalize practice opportunities counts to the 0–1 range, in order for the optimizer not to ignore some of the parameters. Other strategies such as log-normalization or capping of the practice opportunities counts can be employed instead, or in the combination with the 0–1 normalization.

If there are too few failures, the models might overfocus on the dominant successful attempts. One possible remedy is to redefine success to obtain a more balanced distribution of classes [5]. There are several performance measures that could be considered for introductory programming, using either response time or execution count [9].

Another remedy is to undersample successful attempts or oversample the failed ones. Resampling can also be used to reduce other biases in data; for example, balancing item population to get the same proportions of attempts for each item. If just a few easiest items have most attempts (as is the case in RoboMission), this technique will eliminate the danger of the model neglecting the more difficult items. Similarly, balancing student pathways through items could help to reduce attrition and item ordering biases in the data.

Fitting the model requires the choice of an optimizer, together with a regularization strength (which is in the original model applied only on α_i, but may be helpful to add to other parameters as well). As the standard randomized k-fold cross-validation would lead to predicting past information from the future attempts, a cross-validation scheme that avoids this issue should be used; e.g., using time-aware or student-level splits [27]. Finally, the best model can be chosen according to many reasonable metrics, e.g., RMSE, AUC, precision, recall, F1, or balanced accuracy.

Figure 2 shows that while filtering students with few attempts affects the measured RMSE, it does not change the ordering of the Q-matrices according to the model performance. In the following experiments, we always filter students with less than 10 attempts.

The ordering of Q-matrices is also robust with respect to the evaluation metrics, as shown in Figure 3. However, the ability to distinguish between different models varies across metrics; most notably the AUC metric leads to nearly identical values for the AFM-BG and AFM-ABG models and similarly for AFM-BGT models with different Q-matrices. Also the relative performance compared to the item-average baseline varies: in RMSE and AUC metrics, only the AFM-BGT model outperforms this baseline, while in the balanced accuracy metric, also AFM-BG (with Q1 or Q2) outperforms

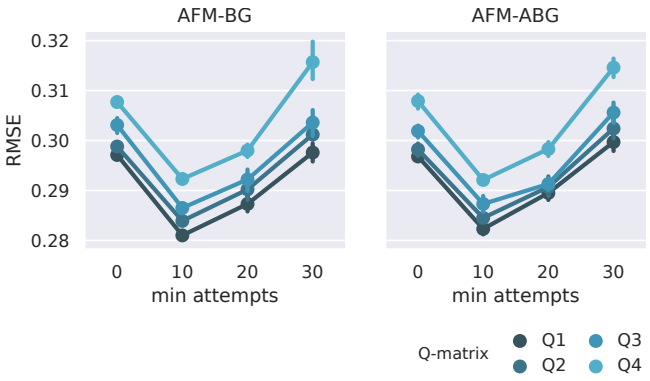

Figure 2: Effect of the choice of threshold for filtering students on the ordering of Q-matrices according to the RMSE of corresponding AFMs. Vertical bars show 95% confidence intervals based on 10 repeated experiments.

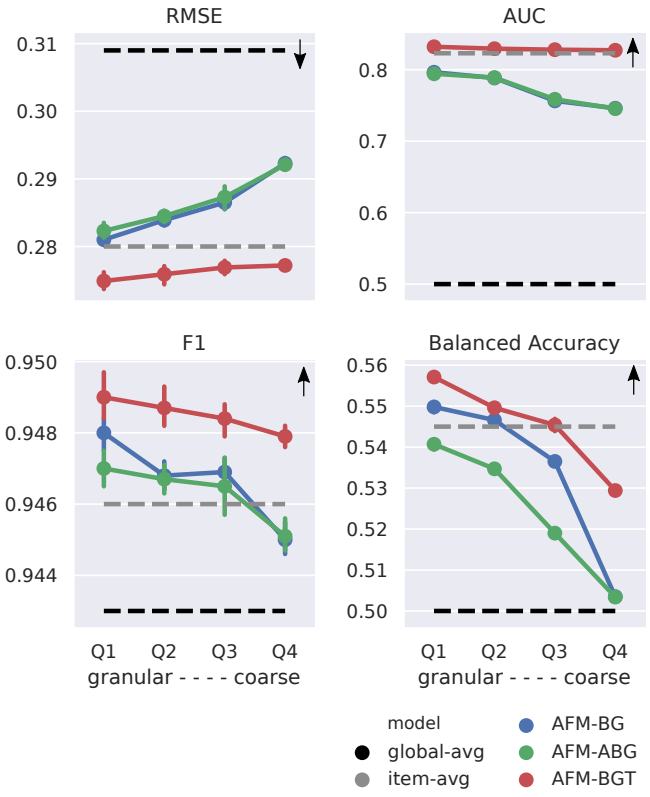

Figure 3: Effect of the choice of evaluation metric on the ordering of Q-matrices according to the performance of corresponding AFMs. The arrows indicate optimization direction—for RMSE, the lower is better, for AUC, F1, and Balanced Accuracy, the higher is better. In the AUC plot, the blue line (AFM-BG model) is hidden below the green line (AFM-ABG model). Vertical bars show 95% confidence intervals based on 10 repeated experiments.

it, and all the AFM variants outperform the baseline in the F1 metric (with Q1, Q2, and Q3 matrices).

In the above experiments, we use 5-fold student-level cross-validation, which means that all attempts of a single student are either in the training set or the test set, reflecting the requirement for generalization to new, unseen students. As expected, randomized k-fold cross-validation, but also the version with time-aware splits, lead to too optimistic RMSE compared to the student-level cross-validation scheme. However, even this methodology choice does not influence the ordering of Q-matrices on our data (Figure 4).

To summarize, ordering of AFMs is robust with respect to the filtering of students with few attempts, chosen evaluation metric, and cross-validation scheme, at least on our data. This does not mean that these choices necessarily does not matter at all—they may still influence other aspects, such as predictions on different subpopulations, or the values of fitted AFM parameters, which are important for conclusions about the learning domain. A deeper analysis of these aspects is left for future work.

5.4 Q-matrix Granularity and Item Difficulty

On the programming data, AFMs achieved lower RMSE than the global success rate baseline, but not higher than the item success rate baseline (Figure 3). The AFM assumes that the items with identical rows in the Q-matrix have the same difficulty, which is violated in our data. An undesirable consequence is that the most fine-grained Q-matrix minimizes the RMSE, even if it is a less truthful picture of reality, just because more item-related parameters allow to predict more closely at least the item success rates. Dangerously, even completely made-up concepts can help the AFM to more closely emulate the item success rate predictor. Consider the following experiment: select a random number of new concepts (1–10) and assign each to 10% of randomly selected items. In our data, adding these made-up concepts results in slightly lower RMSE on average, with improvement in 60% of experiments (mean difference of −0.001). Without considering the violated assumptions of the model, such a result could lead to a false conclusion that the made-up concepts are true and reflect the learning domain.

As a solution, the AFM can be combined with the item average model. Alternatively, the Q-matrix can be extended by adding a special concept for each item. This results in an improvement over the item success rate model, which means that the concepts bring some new information (the red line in Figure 3). Also, while the most granular Q-matrix (Q1) still results in the lowest RMSE, the differences between the other matrices is now lower (the difference in RMSE between Q1 and Q4 drops from 0.011 to 0.002).

On the other hand, adding student priors did not result in a decrease of RMSE on our data and even led to worse

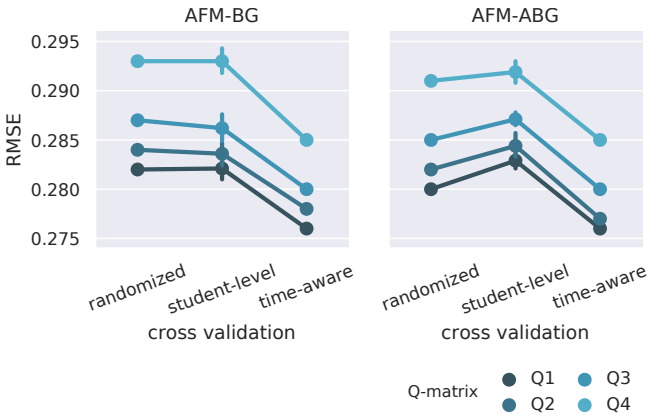

Figure 4: Effect of the choice of cross validation setting on the ordering of Q-matrices according to the performance of corresponding AFMs. Vertical bars show 95% confidence intervals based on 10 repeated experiments.

performance on some other metrics (the green line in Figure 3). All three AFM variants lead to the same ordering of Q-matrices.

5.5 Learning Curves

Figure 5 shows learning curves using AFM with the most granular Q-matrix (AFM-BG-Q1 model) for four concepts in RoboMission. As the students generally progress from easier to more difficult items (item ordering bias), the empirical learning curves are increasing. Consequently, marginal learning curves are expected to increase as well, in order for the model to fit the observed data. On the other hand, if the model would not be susceptible to the item ordering bias, the completed and idealized learning curves should be decreasing, because the points on each curve are computed from the same population of items.

AFM is, however, susceptible to the item ordering bias. For this reason, even the completed and idealized learning curves are either increasing or approximately constant. This illustrates that if the assumptions of AFM are not met, model parameters try to compensate for the violations and, as a result, they do not retain their original interpretation as the initial concept easinesses and learning rates. Interpreting the model parameters or visualized learning curves in this way would lead to false conclusions.

6 CONCLUSIONS

Our explorations of the Additive Factors Model show that the current state-of-the-art of the application of the model is not careful enough. The model makes many assumptions, which are clearly not completely satisfied in real applications. This is necessary as every model makes assumptions. However, it is important to be aware of these assumptions, discuss them explicitly, and check whether they are reasonably satisfied. Our experiments show that when assumptions are violated,

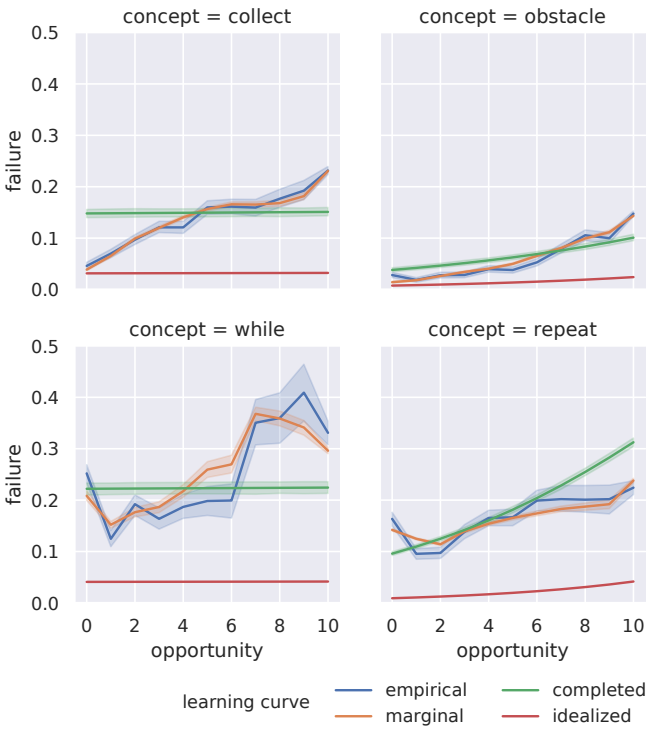

Figure 5: Learning curves for two game concepts (top row) and two programming concepts (bottom row) using AFM fitted on the data from RoboMission.

an analysis based on the model can easily lead to misleading conclusions. In our case of items of different difficulty, AFM predictions can be improved even by the addition of a random knowledge component.

We would like to clarify that we do not claim that previous studies based on the Additive Factors Model present misleading results. Interpretations obtained in some of these studies have been used to redesign a tutoring system and the improved version was rigorously tested in an experiment (e.g., [17]). But we believe that the methods used in the current research could potentially lead to misleading results and thus that greater care should be taken. Specifically, many AFM studies compare only several variants of the model with different Q-matrices. A basic step to improving the state-of-the-art is to include in the comparison also basic baselines like "item average". When predictions of a fitted model are worse than such simple baseline (as happened in some of our cases), we should check which model assumptions are being violated.

Our explorations also show difficulties in replicating previous work. Many important details are not explicitly described, e.g., the details of parameter fitting procedures, the details of the used experimental methodology (e.g., the division of data into a training set and a testing set, treatment of student parameters α in the testing set), or the exact method used for computation of learning curves. We have explored and

described some of these details; some of them can nontrivially influence the results of model comparison.

For our analysis, we have explored data about introductory programming. For our data, AFM does not provide satisfactory fit and insights. One important simplifying AFM assumption, which is not satisfied in this domain, is that the relevant information about student performance is given by the observed binary outcome (solved/unsolved). In programming exercises, other aspects of student performance (specifically problem solving times) contains important information for modeling. AFM could be rather naturally modified to model problem solving time (leading to a model analogical to multidimensional model of problem solving times reported in [28]). We consider this an interesting direction for further research.

REFERENCES

[1] Vincent Aleven and Kenneth R Koedinger. 2013. Knowledge Component (KC) Approaches to Learner Modeling. In *Design Recommendations for Intelligent Tutoring Systems*, Robert A Sottilare, Arthur Graesser, Xiangen Hu, and Heather Holden (Eds.). Vol. 1. US Army Research Laboratory, 165–182.

[2] Hao Cen. 2009. *Generalized learning factors analysis: improving cognitive models with machine learning*. Ph.D. Dissertation. Carnegie Mellon University.

[3] Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *Proc. of Intelligent Tutoring Systems*. Springer, 164–175.

[4] Min Chi, Kenneth Koedinger, Geoff Gordon, Pamela Jordan, and Kurt VanLehn. 2011. Instructional factors analysis. In *Proc. of Educational Data Mining*.

[5] Guillaume Durand, Cyril Goutte, Nabil Belacel, Yassine Bouslimani, and Serge Leger. 2017. *Review, computation and application of the Additive Factor Model (AFM)*. Technical Report. Tech. Report 23002483. National Research Council Canada.

[6] Guillaume Durand, Cyril Goutte, and Serge Léger. 2018. Standard Error Considerations on AFM Parameters. *International Educational Data Mining Society* (2018).

[7] Tomáš Effenberger. 2019. Blockly Programming Dataset. In *3rd Educational Data Mining in Computer Science Education (CSEDM) Workshop*.

[8] Tomáš Effenberger and Radek Pelánek. 2018. Towards making block-based programming activities adaptive. In *Proceedings of Learning at Scale*. ACM, 13.

[9] Tomáš Effenberger and Radek Pelánek. 2019. Measuring Students Performance on Programming Tasks. In *Proceedings of Learning at Scale*. ACM.

[10] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[11] April Galyardt and Ilya Goldin. 2015. Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining* 7, 2 (2015), 83–108.

[12] Yue Gong, Joseph E Beck, and Neil T Heffernan. 2011. How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Journal of Artificial Intelligence in Education* 21, 1-2 (2011), 27–46.

[13] Cyril Goutte, Guillaume Durand, and Serge Léger. 2018. On the Learning Curve Attrition Bias in Additive Factor Modeling. In *International Conference on Artificial Intelligence in Education*. Springer, 109–113.

[14] Tanja Käser, Kenneth R Koedinger, and Markus Gross. 2014. Different parameters - same prediction: An analysis of learning curves. In *Proc. of Educational Data Mining*. 52–59.

[15] Kenneth R Koedinger, Ryan SJd Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. 2010. A data repository for the EDM community: The PSLC DataShop. *Handbook of educational data mining* 43 (2010), 43–56.

[16] Kenneth R Koedinger, Elizabeth A McLaughlin, and John C Stamper. 2012. Automated Student Model Improvement.. In *Proc. of Educational Data Mining*. ERIC.

[17] Kenneth R Koedinger, John C Stamper, Elizabeth A McLaughlin, and Tristan Nixon. 2013. Using data-driven discovery of better student models to improve student learning. In *Proc. of Artificial intelligence in education*. Springer, 421–430.

[18] Ran Liu and Kenneth R Koedinger. 2015. Variations in Learning Rate: Student Classification Based on Systematic Residual Error Patterns across Practice Opportunities.. In *Proc. of Educational data mining*. ERIC.

[19] Ran Liu and Kenneth R Koedinger. 2017. Going beyond better data prediction to create explanatory models of educational data. *The Handbook of Learning Analytics* (2017), 69–76.

[20] Ran Liu, Kenneth R Koedinger, and Elizabeth A McLaughlin. 2014. Interpreting Model Discovery and Testing Generalization to a New Dataset. In *Proc. of Educational Data Mining*. 107–113.

[21] Yanjin Long, Kenneth Holstein, and Vincent Aleven. 2018. What exactly do students learn when they practice equation solving?: refining knowledge components with the additive factors model. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*. ACM, 399–408.

[22] Christoph Molnar. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.

[23] Huy Nguyen, Yeyu Wang, John Stamper, and Bruce M McLaren. 2019. Using Knowledge Component Modeling to Increase Domain Understanding in a Digital Learning Game. In *Proc. of Educational Data Mining*.

[24] Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. 2009. Performance Factors Analysis-A New Alternative to Knowledge Tracing.. In *Proc. of Artificial Intelligence in Education*. IOS Press, 531–538.

[25] Radek Pelánek. 2015. Metrics for Evaluation of Student Models. *Journal of Educational Data Mining* 7, 2 (2015), 1–19.

[26] Radek Pelánek. 2017. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction* 27, 3 (2017), 313–350.

[27] Radek Pelánek. 2018. The details matter: methodological nuances in the evaluation of student models. *User Modeling and User-Adapted Interaction* 28 (2018), 207–235. Issue 3.

[28] Radek Pelánek and Petr Jarušek. 2015. Student Modeling Based on Problem Solving Times. *International Journal of Artificial Intelligence in Education* 25, 4 (2015), 493–519. https://doi.org/10.1007/s40593-015-0048-x

[29] Kelly Rivers, Erik Harpstead, and Kenneth R Koedinger. 2016. Learning curve analysis for programming: Which concepts do students struggle with?. In *ICER*. 143–151.