7-2011

Instructional Factors Analysis: A Cognitive Model For Multiple Instructional Interventions

Min Chi
Stanford University

Kenneth R. Koedinger
Carnegie Mellon University, koedinger@cmu.edu

Geoffrey J. Gordon
Carnegie Mellon University, ggordon@cs.cmu.edu

Pamela Jordon
University of Pittsburgh

Kurt VanLahn
Arizona State University

Published In
Proceedings of the 4th International Conference on Educational Data Mining, 61-70.

Instructional Factors Analysis: A Cognitive Model For Multiple Instructional Interventions

Min Chi, Stanford University, CA USA
Kenneth Koedinger, Carnegie Mellon University, PA USA
Geoff Gordon, Carnegie Mellon University, PA USA
Pamela Jordan, University of Pittsburgh, PA USA
Kurt VanLehn, Arizona State University, AZ USA

In this paper, we proposed a new cognitive modeling approach: Instructional Factors Analysis Model (IFM). It belongs to a class of Knowledge-Component-based cognitive models. More specifically, IFM is targeted for modeling student's performance when multiple types of instructional interventions are involved and some of them may not generate a direct observation of students' performance. We compared IFM to two other pre-existing cognitive models: Additive Factor Models (AFMs) and Performance Factor Models (PFMs). The three methods differ mainly on how a student's previous experience on a Knowledge Component is counted into multiple categories. Among the three models, instructional interventions without immediate direct observations can be easily incorporate into the AFM and IFM models. Therefore, they are further compared on two important tasks—unseen student prediction and unseen step prediction—and to determine whether the extra flexibility afforded by additional parameters leads to better models, or just to over fitting. Our results suggested that, for datasets involving multiple types of learning interventions, dividing student learning opportunities into multiple categories is beneficial in that IFM out-performed both AFM and PFM models on various tasks. However, the relative performance of the IFM models depends on the specific prediction task; so, experimenters facing a novel task should engage in some measure of model selection.

1. INTRODUCTION

For many existing Intelligent Tutoring Systems (ITSs), the system-student interactions can be viewed as a sequence of steps [VanLehn 2006]. Most ITSs are student-driven. That is, at each time point the system elicits the next step from students, sometimes with a prompt, but often without any prompting (e.g., in a free form equation entry window where each equation is a step). When a student enters an attempt on a step, the ITS records whether it is a success or failure *without the tutor's assistance* and may give feedbacks and/or hints based on the entry. Students' first attempt records on each step are then collected for student modeling. Often times in ITSs, completion of a single step requires students to apply multiple Knowledge Components. A *Knowledge Component (KC)* is: "a generalization of everyday terms like concept, principle, fact, or skill, and cognitive science terms like schema, production rule, misconception, or facet" [VanLehn et al. 2007]. They are the atomic units of knowledge. Generally speaking, students' modeling on conjunctive-KC steps are more difficult than that on steps that require a single KC.

The three most common student modeling methods are: **Knowledge Tracing (KT)** [Corbett and Anderson 1995], **Additive Factor Models (AFM)** [Cen et al. 2006; 2008], and **Performance Factor Models (PFM)** [Pavlik et al. 2009]. When performing student modeling we seek to construct a cognitive model based upon these observed behaviors and to apply the model to make predictions. Generally speaking, we are interested in three types of predictions: type 1 is about how *unseen students* will perform on the observed steps same as those in the observed dataset; type 2 is about how the same students seen in the observed data will perform on *unseen steps*; and type 3 is about how unseen students will perform on unseen steps, that is, *both*. For the present purposes we classifie students or steps that appear in the observed training data

as seen and those that appear only in the unobserved test data as unseen. In this paper we will examine prediction types 1 and 2 and leave type 3 for future work.

Previously KT and PFM have been directly compared both on datasets involved single-KC steps [Pavlik et al. 2009] and those involved conjunctive-KC steps[Gong et al. 2010]. Results have shown that PFM is as good or better than KT for prediction tasks under Bayesian Information Criteria (BIC) [Schwarz 1978] in [Pavlik et al. 2009] or using Mean Squared Error (MSE) as criteria in [Gong et al. 2010]. For both BIC and MSE, the lower the value, the better.

While PFM and KT have been compared on datasets involved conjunctive-KC step, prior applications of AFM and PFM have mainly been with single-KC steps and indicated no clear winner. More specifically, while AFM is marginally superior to PFM in that the former has lower BIC and cross-validation Mean Absolute Deviance (MAD) scores in [Pavlik et al. 2009], PFM performed better than AFM under MAD scores in [Pavlik et al. 2011]. For MAD, same as MSE, the lower the value, the better. On the other hand, previous research have shown that AFM can, at least in some cases, do a fine job in modeling conjunctive KCs [Cen et al. 2008]. Therefore, in this paper we will compare AFM and PFM directly on a dataset involving many conjunctive-KC steps.

Moreover, most prior research on cognitive modelings was conducted on datasets collected from classical student-driven ITSs. Some ITSs, however, are not always student-driven in that they may involve other instructional interventions that do not generate direct observations on student's performance. The dataset used in this paper, for example, was collected from a tutor that, at each step chose to elicit the next step information from students or to tell them the next step. In our view these tell steps should also be counted as a type of Learning Opportunity (LO) as they do provide some guidance to students. Yet on the other hand, these steps do not allow us to directly observe students' performance. KT model is designed mainly for student-driven ITSs in that its parameters are directly learned from the sequences of student's performance (right or wrong) on each step. When there are multiple instructional interventions and some of them do not generate direct observations, it is not very clear how to incorporate these interventions directly into conventional KT models. Therefore, in this paper we are mainly interested in comparing AFM and PFM.

Our dataset was collected from an ITS that can either *elicit* the next step from the student or *tell* them directly. Incorporating tell steps into AFM model is relatively easy in that tells can be directly added to total LO counts. The PFM, however, uses student's prior performance counts, the success or failure, in the equation. Since tells do not generate any observed performance, it is hard to include them in the PFM. Therefore, we elected to add a new feature to represent instructional interventions such as tells. As shown later, the new model can be easily modified for modeling datasets with multiple instructional interventions and thus it is named as Instructional Factors Analysis Model (IFM).

To summarize, in this paper we will compare three models, AFM, PFM and IFM, on a dataset involving many conjunctive-KC steps and multiple instructional interventions. Previous research has typically focused on how well the models fit the observed data. In the following, we also investigated how well they perform at making the predictions of unseen students' performance on seen steps (type 1) and seen students' performance on unseen steps (type 2). Before describing our general methods in details we will first describe the three models.

2. THREE MODELS: AFM, PFM, AND IFM

All three models, AFM, PFM, and IFM, use a *Q-matrix* to represent the relationship between individual steps and KCs. Q-matrices are typically encoded as a binary 2-dimensional matrix with rows representing KCs and columns representing Steps. If a given cell $Q_{kj} = 1$, then step j is an application of KC k. Previous researchers have focused on the task of generating or tuning Q-matrices based upon a dataset [Barnes 2005; Tatsuoka 1983]. For the present work we employed a static Q-matrix for all our experiments. Equations 1,

2, and 3 present the core of each model. Below the equations are the detailed descriptions of each term used in the three equations.

The central idea of AFM was originally proposed by [Draney et al. 1995] and introduced into ITS field by [Cen et al. 2006; 2008]. Equation 1 shows that AFM defines the log-odds of a student i completing a step j correctly to be a linear function of several covariates. Here p_{ij} is a student i's probability of completing a step j correctly, N_{ik} is the prior LO counts. AFM models contain three types of parameters: student parameters θ_i, KC (or skill) parameters β_k, and learning rates γ_k. While AFM is sensitive to the frequency of prior practice, it assumes that all students accumulate knowledge in the same manner and ignores the correctness of their individual responses.

PFM, by contrast, was proposed by [Pavlik et al. 2009] by taking the correctness of individual responses into account. It can be seen as a combination of learning decomposition [Beck and Mostow 2008] and AFM. Equation 2 expresses a student i's log-odds of completing a step j correctly based upon performance features such as S_{ik} (the number of times student i has previously practiced successfully relevant KC k) and F_{ik} (the number of times student i has previously practiced unsuccessfully relevant KC k). PFM may also include student parameters such as θ_i and skill parameters, such as β_k. Additionally, PFM employs parameters to represent the benefit of students' prior successful applications of the skill μ_k and the benefit of prior previous failures ρ_k.

While PFM was originally proposed without a θ_i, it is possible to include or exclude these student parameters from either PFM or AFM. In prior work, Corbett et al. noted that models which tracked learning variability on a per-subject basis, such as with θ outperform models that do not [Corbett and Anderson 1995]. Pavlik [Pavlik et al. 2009] further noted that the full AFM model seemed to outperform PFM without θ which in turn outperformed AFM without θ. Pavlik et al. also hypothesized that PFM with θ would outperform the other models and they investigated it in their recent work. In this study, our analysis showed that prediction is better with student parameters, especially for AFM models, thus we include θ_i in our versions of both AFM and PFM.

From PFM, IFM can be seen as adding a new feature to represent the tells together with the success or failure counts, shown in Equation 3. Equation 3 expresses a student i's log-odds of completing a step j correctly based upon performance features including S_{ik}, F_{ik}, T_{ik} (the number of times student i has previously got told on relevant KC k). IFM also includes student parameters θ_i, skill parameters β_k, μ_k, ρ_k, and the benefit of prior previous tells ν_k.

$$\textbf{AFM:} \quad \ln \frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}(\gamma_k N_{ik}) \tag{1}$$

$$\textbf{PFM:} \quad \ln \frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}(\mu_k S_{ik} + \rho_k F_{ik}) \tag{2}$$

$$\textbf{IFM:} \quad \ln \frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}(\mu_k S_{ik} + \rho_k F_{ik} + \nu_k T_{ik}) \tag{3}$$

Where:

- i. represents a student i.
- j. represents a step j.
- k. represents a skill or KC k.
- p_{ij}. is the probability that student i would be correct on step j.
- θ_i. is the coefficient for proficiency of student i.
- β_j. is coefficient for difficulty of the skill or KC k.

Q_{kj}. is the Q-matrix cell for step j using skill k.

γ_k. is the coefficient for the learning rate of skill k (AFM only);

N_{ik}. is the number of practice opportunities student i has had on the skill k (AFM only);

μ_k. is the coefficient for the benefit of previous successes on skill k (PFM & IFM);

S_{ik}. is the number of prior successes student i has had on the skill k (PFM & IFM);

ρ_k. is the coefficient for the benefit of previous failures on skill k (PFM & IFM);

F_{ik}. is the number of prior failures student i has had on the skill k (PFM & IFM);

ν_k. the coefficient for the benefit of previous tells on skill k (IFM only);

T_{ik}. the # of prior Tells student i has had on the skill k (IFM only);

3. TRAINING DATASET AND EIGHT LEARNING OPPORTUNITY MODES

The original dataset was collected by training 64 students on a natural-language physics tutoring system named Cordillera [VanLehn et al. 2007; Jordan et al. 2007] over a period of four months in 2007. The physics domain contains **eight** primary KCs including the weight law (KC_1), Definition of Kinetic Energy (KC_{20}), Gravitational Potential Energy (KC_{21}), and so on. All participants began with a standard pretest followed by training 7 physics problems on Cordillera and then a post-test. The pre- and post-tests are identical in that they both have the same 33 test items. The tests were given online and consisted of both multiple-choice and open-ended questions. Open-ended questions required the students to derive an answer by applying one or multiple KCs.

In this study, our training dataset comprises 19301 data points resulted from 64 students solving 7 training problems on Cordillera. Each student completed around 300 training problem steps. Note that the training dataset does not include the pre- or posttest. In other words, a data point in our training dataset is either the first attempt by a student on an elicit step or a system tell during his/her training on Cordillera only.

There are two types of steps in Cordillera. The *primary steps* are necessary problem-solving and conceptual discussion steps. The *justification steps*, on the other hand, are optional steps that occur when students are asked to justify the primary step they have just completed. The primary steps are designed to move the solution process forward while the justification steps are designed to help the students engage with the domain knowledge in a deeper way. When collecting our dataset the Cordillera system decided whether to elicit or tell each step randomly. Thus, we have two types of LOs: *elicit* and *tell* for the primary steps; and *self-explain* or *explain* for the justifications.

Figure 1 shows a pair of sample dialogues taken from the cordillera system for the same series of primary steps with the same domain content. In dialogue (1.a) the system elects to elicit the students' answer (steps 2- 3), while in dialogue (1.b) the system chooses to tell the student the answer (steps 2). Similarly in Figure 2 we present a similar comparison between a pair of self-explain and explain dialogues. As before both dialogues cover the same domain content. In dialogue (2.a) the system asks the student to self-explain their answer to the question in qualitative terms (steps 3-4). In dialogue (2.b) they are provided with a short qualitative explanation (step 3).

For the primary steps, the average number of decisions ranges from 2.5 for KC_1 to 31 for KC_{20} for elicit steps and ranges from 4 for KC_1 to 47 for KC_{20} for the tell steps. Compared with primary steps, justification steps are significantly less frequent in that the average number ranges from 0 for KC_1 to 3.5 for KC_{21} for *self-explain* steps and ranges from 0 for KC_1 to 2.5 for KC_{20} for explain steps.

In most of the prior research that we have surveyed the authors mainly used datasets involving quantitative problem-solving steps similar to the primary steps in Cordillera. So when counting LOs in this study, we first need to decide whether or not to include justification steps. Additionally, most previous research did not include students' performance on the pretest when counting LOs. While this makes some sense as students

(1.a) Elicit Version

(1) **T:** So let's start with determining the value of v1.
(2) **T:** Which principle will help you calculate the rock's instantaneous magnitude of velocity at T1? **{ELICIT}**
(3) **S:** definition of kinetic energy
(4) **T:** Let me just write the equation for you: KE1 = 0.5*m*v1^2.

(1.b) Tell Version

(1) **T:** So let's start with determining the value of v1.
(2) **T:** To calculate the rocks instantaneous magnitude of velocity at T1, we will apply the definition of kinetic energy. **{TELL}**
(3) **T:** Let me just write the equation for you: KE1 = 0.5*m*v1^2.

Fig. 1. Learning Opportunities on Problem-Solving Steps: Elicit vs. Tell

(2.a) Self-Explain

(1) **T:**Can we infer the direction of the velocity of the rock at T1 from the rock's kinetic energy at T1? **{ELICIT}**
(2) **S:**nope.
(3) **T:**Excellent! Please explain why. **{Self-Explain}**
(4) **S:**Only the magnitude of the velocity and not the direction of it is part of the definition of kinetic energy.
(5) **T:**Excellent! Now that we know v1, $\cdots$

(2.b) Explain

(1) **T:**Can we infer the direction of the velocity of the rock at T1 from the rock's kinetic energy at T1? **{ELICIT}**
(2) **S:**nope.
(3) **T:**Excellent! This is because the kinetic energy only depends on mass and the magnitude of velocity, not the direction of velocity.**{Explain}**
(4) **T:**Now that we know v1, $\cdots$

Fig. 2. SelfExplain vs. Explain

receive no feedback indicating their successes or failures during the test, it is still the case that they do practice their skills. Therefore, secondly we need to decide whether or not to include student's pretest performance in the LO counts.

In order to explore how different choices of LOs would impact different cognitive models, we defined four ways to count the LOs. In the *primary mode* we count only the primary steps within the ITS. In *pretest-primary* we count the primary mode steps plus the pretest (each test item is treated as one step for training). *Primary-Justify* mode counts the primary and justification steps within the ITS alone. And finally the *overall* mode counts all steps in both the pretest and ITS training.

Note that using different modes of LOs neither changes the size of the training dataset which is generated along students' logs when training on Cordillera nor changes the number of parameters to be fit. Using pretest in the LO count means that various LOs do not start with 0 for the *pretest-primary* and *overall*

modes but are based on the frequency of KC appearances (and, in the case of PFM, the accuracy) in the pretest. For example, if a KC_{20} is tested 20 times in the pretest and a student was correct 5 times and wrong 15 times, then the student's LOs on KC_{20} for pretest-primary and *overall* mode would start with $LO = 20, Success = 5, Fail = 15, Tell = 0$. For *Primary* and *Primary-Justify* modes, all LOs start with 0.

Coupled with this variation we can also count LOs additively or logarithmically. Using logarithmic count is inspired by the power law relationship between measures of performance (reaction time or error rate) and the amount of practice [Newell and Rosenbloom 1981]. But others [Heathcote et al. 2000] have argued that the relationship is an exponential, which corresponds to additive counting. To summarize, we have {Primary, Pretest-Primary, Primary-Justify, Overall} × {count, ln(count)}, a total of eight LO modes.

4. RESULTS

Two measures of quality, the Bayesian Information Criteria (BIC) and the cross-validation Root Mean Squared Error (RMSE), are used to evaluate how well various instantiated models perform. For both BIC and cross-validation RMSE, the lower the value, the better. BIC [Schwarz 1978] is a criterion for model selection among a class of parametric models with different numbers of parameters. In prior research on the evaluation and comparison of different cognitive models [Cen et al. 2006; Pavlik et al. 2009; Gong et al. 2010] the authors used BIC as a measure of success. In machine learning, however, it is conventional to use the cross-validation RMSE, which is a more interpretable metric and, we believe, a more robust measure. For the purposes of this paper, we will report both BIC and RMSE.

4.1 AFM, PFM, vs. IFM.

First, we will investigate whether considering Tell and Explains into the LOs is beneficial. In traditional cognitive modeling the focus is solely on steps where the student's performance is observed. In the context of Cordillera that means counting only the elicit and self-explain steps as both require students to apply their knowledge without support and their performance can be directly evaluated. For AFM models, we thus compared the AFM algorithms shown in equation 1 by either including Tells and Explains into N_{ik} or by excluding them out of N_{ik}. The two resulted models are referred as AFM-Tell and AFM+Tell respectively. Therefore, in this section we compared four models: AFM-Tell, AFM+Tell, PFM and IFM across eight LO modes.

For each of the four models, its corresponding **count** LOs on corresponding {Primary, Pretest-Primary, Primary-Justify, Overall} modes are defined in Table I. For example, the IFM has three LO counts: prior success S_{ik}, prior failures F_{ik}, and prior tells T_{ik}. Under the Primary-Justify mode (shown in the left bottom of the table), S_{ik} = Success in (Elicit + Self-Explain) on the KC k, F_{ik} = prior failure in (Elicit + Self-Explain) on the KC k, and T_{ik} = prior tells and explains on the KC k. Once the count mode is defined, the corresponding Ln(Count) mode is simply taking each count logarithmically. For example, under {Primary-Justify, Ln(Count)} mode, we have S_{ik} = ln[Success in (Elicit + Self-Explain) on KC k], F_{ik} = ln[prior failure in (Elicit + Self-Explain) on KC k], and T_{ik} = ln[prior tells and explains on the KC k].

For each model on each mode, we carried out a 10-fold cross-validation. Such procedure resulted in 8 (modes) × 4 (models) = 32 BIC values and CV RMSE values. Table II shows the comparisons among the four models when using {Primary-Justify, Count} and {Primary-Justify, Ln(Count)} LO modes respectively. It shows that across both modes, the IFM is more accurate (both lower BIC and RMSE) than the PFM; similarly, the latter is more accurate than AFM+Tell and AFM-Tell. However, it is harder to compare AFM-Tell and AFM+Tell. For example, on {Primary-Justify, Count} mode, although AFM-Tell has lower BIC than AFM+Tell 9037 vs. 9058, the latter has lower RMSE than the former: 4.456E-01 vs. 4.459E-01. So on both {Primary-Justify, Count} and {Primary-Justify, Ln(Count)} modes, we have IFM > PFM > AFM+Tell, AFM-Tell. Such pattern is consistence across all eight modes.

Table I. {Primary, Pretest-Primary, Primary-Justify, Overall} Learning Opportunity Modes

		Primary	**Pretest-Primary**
AFM-Tell	N_{ik}	Elicit	Pretest+Elicit
AFM+Tell	N_{ik}	Elicit+Tell	Pretest+Elicit+Tell
PFM	S_{ik}	Success(Elicit)	Success in (Pretest + Elicit)
	F_{ik}	Failure(Elicit)	Failure in (Pretest + Elicit)
IFM	S_{ik}	Success(Elicit)	Success in (Pretest + Elicit)
	F_{ik}	Failure(Elicit)	Failure in (Pretest + Elicit)
	T_{ik}	Tell	Tell
		Primary-Justify	**Overall**
AFM-Tell	N_{ik}	Elicit + SelfExplain	Pretest+ Elicit+SelfExplain
AFM+Tell	N_{ik}	Elicit+Tell + SelfExplain +Explain	Pretest+ Elicit+Tell + SelfExplain+Explain
PFM	S_{ik}	Success in (Elicit + Self-Explain)	Success in (Pretest+ Elicit + Self-Explain)
	F_{ik}	Failure in (Elicit + Self-Explain)	Failure in (Pretest+ Elicit + Self-Explain)
IFM	S_{ik}	Success in (Elicit + Self-Explain)	Success in (Pretest+ Elicit + Self-Explain)
	F_{ik}	Failure in (Elicit + Self-Explain)	Failure in (Pretest+ Elicit + Self-Explain)
	T_{ik}	Tell+ Explain	Tell+ Explain

Table II. Compare AFM-Tell, AFM+Tell, PFM and IFM on {Primary-Justify, Count} and {Primary-Justify, Ln(Count)} mode

	{Primary-Justify, Count}		{Primary-Justify, Ln(Count)}	
Model	BIC	10-fold RMSE	BIC	10-fold RMSE
AFM-Tell	9037	4.460E-01	9037	4.459E-01
AFM+Tell	9117	4.470E-01	9058	4.456E-01
PFM	8474	4.235E-01	8461	4.236E-01
IFM	8347	4.217E-01	8321	4.211E-01

In order to compare the performance among four models, Wilcoxon Signed Ranks Tests were conducted on resulted BICs and RMSEs. Results showed that IFM significantly outperformed the PFMs across eight modes: $Z = -2.52$, $p = 0.012$ for both BIC and cross-validation RMSE. Similarly, it was shown that across all eight modes IFM beat corresponding AFM-Tell across eight modes significantly on both BIC and RMSE: $Z = -2.52$, $p = 0.012$. Similar results were found between IFM and AFM+Tell in that the former out-performed the latter across eight modes significantly on both BIC and RMSE: $Z = -2.52$, $p = 0.012$.

Comparisons between PFM and AFM-Tell and AFM+Tell showed that PFM beats corresponding AFM-Tell across eight modes significantly on both BIC and RMSE: $Z = -2.52$, $p = 0.012$; and PFM also beat AFM+Tell significantly on both BIC and RMSE: $Z = -2.52$, $p = 0.012$. Finally, comparisons between AFM-Tell and AFM+Tell showed that adding Tells and Explains into LOs did not statistically significantly improve the BIC and RMSE of the corresponding AFM model: $Z = -0.28$, $p = 0.78$ for BIC and $Z = -1.35$, $p = 0.18$ for RMSE respectively. Therefore, our overall results suggested: IFM > PFM > AFM-Tell, AFM+Tell.

Next, we investigated which way of counting LOs is better, using logarithmic or additive tabulation? Wilcoxon Signed Ranks Tests were conducted on comparing the BIC and RMSE of the performances when using Count versus using Ln(Count) on the same model and mode. Results showed using Ln(Count) performed significantly better than using Count: $Z = -2.27$, $p = 0.008$ for BIC and $Z = -2.33$, $p = 0.02$ for RMSE respectively. This analysis is interesting in relation to a long-standing debate about whether the learning curve is exponential (like additive tabulation) or a power law (logarithmic tabulation) [Heathcote et al. 2000]. Our results appear to favor the power law.

Next, we investigated the impact of four LO modes. The BICs and RMSEs were compared among the {Primary, Pretest-Primary, Primary-Justify, Overall} modes regardless of Count and Ln(Count). A pairwise comparisons on Wilcoxon Signed Ranks Tests showed that the {Primary-Justify} modes generated signifi-

cantly better models than using {Primary} modes $Z = -2.1$, $p = 0.036$; the {Primary} modes generated better models than using {Pretest-Primary} and {Overall} $Z = -2.27$, $p = 0.018$ and $Z = -2.521$, $p = 0.012$ respectively. While no significant difference was found between {Pretest-Primary} and {Overall} modes. Similar results was found on RMSE. Therefore, it suggested that adding justification steps into LOs is beneficial in that Primary-Justify mode beats Primary; however, adding pretest into the LOs did not produce better models and it may even have resulted worse models: the benefit of adding justification steps into LOs was seemingly washed out by including pretest in the LOs in that {Overall} modes generate worse models than {Primary-Justify} and {Primary}.

To summarize, for modeling the training data, applying IFM model and using {Primary-Justify, Ln(Count)} as LOs generated the best fitting model. Additionally, comparisons among the IFM, PFM, AFM-Tell,and AFM+Tell showed that IFM > PFM > AFM-Tell, AFM+Tell. In this paper, our goal is to compare cognitive models on datasets involving multiple types of instructional interventions. As shown above, for AFM the tell steps can be directly added into existing opportunity count N_{ik}; For the PFM model, however, there is no direct way how tells should be incorporated. Therefore, in the following we will mainly compare IFM and AFM+Tell. For the convenient reasons, we will refer to AFM+Tell as AFM.

4.2 IFM vs. AFM for Unseen Student Prediction (Type 1)

Next we compared the AFM and IFM models on the task of unseen student prediction. In order to predict unseen student's performance, Student ID was treated as a random factor in both AFM and IFM models. Here we conducted Leave-one-student-out cross-validation. In other words, 64 students resulted in a 64-fold cross validation. Thus, we have 8 (modes) $\times$ 2 (AFM vs.IFM) BIC values and Cross-Validation RMSE values.

Table III shows the correpsonding BIC and RMSE values of AFM and IFM models using {Primary-Justify, Ln(Count)} mode. Table III shows that IFM generates better prediction models (both lower BIC and RMSE) than AFM and the difference is large. Such pattern is consistence across all eight modes.

Table III. AFM vs. IFM On Unseen Students with Random Effect Student Parameters

Model	BIC	64-fold Cross-Validation RMSE
AFM	8724	4.6144E-01
IFM	7952	4.1661E-01

To compare IFM and AFM across eight modes, Wilcoxon Signed Ranks Tests were conducted on both BICs and cross-validation RMSEs. Consistent with the patterns shown in Table III, results showed that IFM is significant better than AFM across eight modes: $Z = -2.52$, $p = 0.012$ for both BIC and cross-validation RMSE. To summarize, IFM with random student parameter is a better model for predicting unseens students' performances on seen steps than AFM model with random student parameter. The best performance was generated IFM model using {Primary-Justify, Ln(Count)} as LOs.

4.3 AFM vs. IFM for Unseen Step prediction (Type 2).

Finally we compared AFM and IFM models on the task of unseen step prediction. Here we used training dataset and tested each models' prediction using students' post-test performance. For each model on each mode, we carried out a 10-fold cross-validation. Such procedure again resulted in 8×2 BIC values and CV RMSE values.

Table IV shows the results on comparisons for the AFM and IFM models on both {Primary-Justify, Ln(Count)} and {Overall, Ln(Count)} modes. Across the eight LO modes, the performance of AFM reaches its best when using {Primary-Justify, Ln(Count)} mode and IFM reaches its best when using {Overall, Ln(Count)} mode. Table III shows that when using {Primary-Justify, Ln(Count)} mode, the AFM is even

more accurate (both lower BIC and RMSE) than the corresponding IFM model; while when using {Overall, Ln(Count)} LO mode, the IFM is more accurate (both lower BIC and RMSE) than the corresponding AFM. Moreover, the best IFM model, using {Overall, Ln(Count)} LO mode, is still better than the best AFM which using {Primary-Justify, Ln(Count)} LO mode. Thus, cross 8 modes on both AFM and IFM, the best prediction model is still generated by IFM but using {Overall, Ln(Count)} LO mode.

Table IV. AFM vs. IFM On Predicting Post-test Performance by {Primary-Justify, Ln(Count)} and {Overall, Ln(Count)} modes

Mode	Model	BIC	10-fold RMSE
{Primary-Justify, Ln(Count)}	AFM	2414	4.6632E-01
	IFM	2428	4.6791
{Overall, Ln(Count)}	AFM	2443	4.7027E-01
	IFM	2252	4.4529E-01

In order to compare AFM and IFM across eight modes, Wilcoxon Signed Ranks Tests were again conducted on resulted 8×2 BIC and RMSE results. Result showed that IFM is marginally significant better than AFM across eight modes: $Z = -1.68$, $p = 0.093$ for BIC and $Z = -1.82$, $p = 0.069$ for 10-fold CV RMSE respectively. Previously, the best model for fitting the training dataset and type 1 predictions are generated by IFM using {Primary-Justify, Ln(Count)} LOs; on the task of predicting students' posttest performance (type 2), however, the best model is still IFM but using {Overall, Ln(Count)} LO counts. To summarize, the best performance of IFM is better than the best AFM and across the eight LO modes and IFM is marginally better than AFM model on type 2 prediction.

5. CONCLUSION

In this paper we investigated student modeling on a dataset involving multiple instructional interventions. We proposed a cognitive model named IFM. We compared IFM with AFM and PFM on the training dataset. We determined that including non-standard LOs such as tells and explains as a separated parameter is effective in that the IFM models' out-performance PFM, AFM-Tell, and AFM+Tell across all modes; but for AFM modes, simply adding tells into AFM LO counts did not seemingly significantly improved the AFM model's performance. This is probably because AFM gives a same learning rate for different instructional interventions. For example, under the {Primary, Count} mode, the N_{ik} in AFM+Tell model is $Elicit + Tell$. On one KC, KC_{20}, the AFM had: the learning rate $\gamma_k = 0.011462$. By contrast, the corresponding IFM has three parameters: μ_k for benefit of previous successes on skill k; ρ_k is the coefficient for the benefit of previous failures, and ν_k the coefficient for the benefit of previous tells on skill k. For the same KC, the IFM resulted $\mu_k = 0.083397; \rho_k = -0.213746, \nu_k = 0.031982$. The values of the three parameters are quite different from each other, which suggested the the benefit of tells is in the middle of the benefit of success and failure. Such patterns on learned parameters between AFM and IFM showed throughout our analysis. It suggested that rather than using one learning rate parameters for different instructional interventions, it is better to break them into categories and learn seperated parameters.

In order to fully exploring the effectiveness of three models, we further compared them on two prediction tasks – unseen student prediction (type 1) and unseen step prediction (type 2). Our results indicate that the IFM model is significantly better than the AFM model on predicting unseen student's performance on seen steps (type 1) and marginal significant better on predicting seen students' performance on posttest (type 2).

Additionally, we examined the impact of including pretest performance in the LOs as well as qualitative justification steps in the LOs. We found that the Primary-Justify mode seems to be most effective. Generally speaking, models trained with logarithmic tabulation outperformed those trained with additive tabulation

probably because the number of prior LOs counts in this study can be ralatively large. For example, the average number of primary steps (including both elicits and tells) in the training data varies from 6 for KC_1 to 83 for KC_{20}.

Even though IFM model performed the best on modeling the training data on both type 1 and type 2 predictions, its performance is heavily dependent upon the specific prediction task being performed and the way in which the specific LOs were counted. For modeling the training data and type 1 prediction, it is the best to using (Primary-Justify,Ln(Count)) mode; but for type 2 predictions, it was best to include the pretest data as well and thus using(Overall,Ln(Count)) mode for LO counts. Thus we conclude that, for datasets involving multiple learning interventions, IFM is a more robust choice for student and cognitive modeling. However the performance of IFM is heavily dependent upon the specific prediction task being performed and the way in which the specific LOs were counted. Experimenters facing a novel task should engage in some measure of parameter-fitting to determine the best fit.

ACKNOWLEDGMENTS

NSF (#SBE-0836012) and NSF (#0325054) supported this work.

REFERENCES

Barnes, T. 2005. The q-matrix method: Mining student response data for knowledge.

Beck, J. E. and Mostow, J. 2008. How who should practice: Using learning decomposition to evaluate the efficacy of different types of practice for different types of students. See Woolf et al. [2008], 353–362.

Cen, H., Koedinger, K. R., and Junker, B. 2006. Learning factors analysis - a general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems*, M. Ikeda, K. D. Ashley, and T.-W. Chan, Eds. Springer, 164–175.

Cen, H., Koedinger, K. R., and Junker, B. 2008. Comparing two irt models for conjunctive skills. See Woolf et al. [2008], 796–798.

Corbett, A. T. and Anderson, J. R. 1995. Knowledge tracing: Modelling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact. 4,* 4, 253–278.

Draney, K., Pirolli, P., and Wilson, M. 1995. *A Measurement Model for a Complex Cognitive Skill.* Erlbaum, Hillsdale, NJ.

Gong, Y., Beck, J., and Heffernan, N. 2010. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent Tutoring Systems*, V. Aleven, J. Kay, and J. Mostow, Eds. Lecture Notes in Computer Science Series, vol. 6094. Springer Berlin / Heidelberg, 35–44. 10.1007/978-3-642-13388-6_8.

Heathcote, A., Brown, S., and D.J.K., M. 2000. The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin and Review 7,* 2, 185207.

Jordan, P. W., Hall, B., Ringenberg, M., Cue, Y., and Rosé, C. 2007. Tools for authoring a dialogue agent that participates in learning studies. In *AIED*, R. Luckin, K. R. Koedinger, and J. E. Greer, Eds. Frontiers in Artificial Intelligence and Applications Series, vol. 158. IOS Press, Los Angeles, California, USA, 43–50.

Newell, A. and Rosenbloom, P. 1981. *Mechanisms of Skill Acquisition and the Law of Practice.* Erlbaum Hillsdale NJ.

Pavlik, P. I., Cen, H., and Koedinger, K. R. 2009. Performance factors analysis –a new alternative to knowledge tracing. In *Proceeding of the 2009 conference on Artificial Intelligence in Education*. IOS Press, 531–538.

Pavlik, P. I., Yudelson, M., and Koedinger, K. 2011. Using contextual factors analysis to explain transfer of least common multiple skills.

Schwarz, G. E. 1978. Estimating the dimension of a model. *Annals of Statistics. 6,* 2, 461464.

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

VanLehn, K. 2006. The behavior of tutoring systems. *International Journal Artificial Intelligence in Education 16,* 3, 227–265.

VanLehn, K., Jordan, P., and Litman, D. 2007. Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed. In *Proceedings of SLaTE Workshop on Speech and Language Technology in Education ISCA Tutorial and Research Workshop*. 17–20.

Woolf, B. P., Aïmeur, E., Nkambou, R., and Lajoie, S. P., Eds. 2008. *Intelligent Tutoring Systems, 9th International Conference, ITS 2008, Montreal, Canada, June 23-27, 2008, Proceedings*. Lecture Notes in Computer Science Series, vol. 5091. Springer.