User Modeling – a Notoriously Black Art

Michael Yudelson[1], Philip I. Pavlik Jr.[1], and Kenneth R. Koedinger[1]

Human Computer Interaction Institute, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh PA 15213, USA,
ppavlik@andrew.cmu.edu, yudelson@cmu.edu, koedinger@cmu.edu

Abstract. This paper is intended as guidance for those who are familiar with user modeling field but are less fluent in statistical methods. It addresses potential problems with user model selection and evaluation, that are often clear to expert modelers, but are not obvious for others. These problems are frequently a result of a falsely straightforward application of statistics to user modeling (e.g. over-reliance on model fit metrics). In such cases, absolute trust in arguably shallow model accuracy measures could lead to selecting models that are hard-to-interpret, less meaningful, over-fit, and less generalizable. We offer a list of questions to consider in order to avoid these modeling pitfalls. Each of the listed questions is backed by an illustrative example based on the user modeling approach called Performance Factors Analysis (PFA) [9].

Keywords: User modeling, educational data mining, model selection, model complexity, model parsimony

1 Introduction

Fitting a mathematical model of user's behavior to the data is a notoriously black art. While this statement typically is agreed to by expert modelers, it is very difficult to convey exactly what it means to modelers who, while having a fair knowledge of the domain being modeled, do not possess a solid knowledge of statistics. Inexperienced modelers often transfer classroom knowledge of statistics directly into the cognitive domain, which typically results in multiple confusions. Consider, for example, the Akaike information criterion (AIC) or Bayesian information criterion (BIC). These statistics are routinely output by many statistical packages, but over-reliance on these criteria could lead a modeler to making inappropriate inferences, since user modeling data is very infrequently independent, as required by the definitions of AIC and BIC.

In reality, observations are often dependent and are nested by user, by location, or by content items users interact with. Thus, AIC (which gives preference to models with fewer parameters) and BIC (which, in addition, ranks models build using a smaller sample) cannot always account for these nested dependencies. Other frequently used statistics, such as log-likelihood, mean absolute error, r^2, precision, recall, F-measure, or A′ (area under the ROC curve), provide empirical assessments of model's fit, that, although useful for determining whether

each new parameter provides additional explanatory power, offer little support in deciding whether the model makes sense and/or supports a prior theory.

In order to make a better decision on model usefulness, the modeler needs to use other criterions for practical model selection and is faced with a series of questions that need to be posed throughout the modeling process. These questions must be kept foremost in the modeler's mind otherwise the risk exists that the implications of the model will be misinterpreted. These questions are.

1. What factors of the data are used to estimate predictive parameters, and which are used to estimate descriptive parameters?
2. What components of the model are fixed effects of the design, and which are random effects due to the selection from the environment?
3. Is the model complex enough in its identification of parameters with user constructs and user experience?
4. Is the model parsimonious in its identification? Namely, is there little or no polysemy among the parameters?

Attention to the questions in the list above is as important as seeing the effect of changes in the model on the model fit. As we will see in the following user modeling report, it is relatively easy to produce models with better fit (as per, for example, AIC, BIC, r^2, or A$'$ metrics) if these issues are ignored, but these models will be less useful to the modeler and the user modeling community alike. Our goals are similar with those of the authors of [10], for we are arguing against making a compromise when utilizing mentioned metrics, but rather highlighting the cases when their brute-force use truly leads to conflicting conclusions.

The rest of the paper is organized around the items in our checklist in the order their appearance. First, predictive vs. descriptive modeling is addressed. A brief description of our modeling dataset follows. Then, fixed vs. random effects modeling of user-specific parameters is discussed. Finally we talk about model complexity and parsimony.

2 Predictive vs. Descriptive Modeling

Whether the model parameters are estimated in a predictive or descriptive manner is an important aspect of model building, but is often overlooked or ignored. The choice of the way the data is organized and parameters are constructed could have a tangible effect on properties of the model being built. An example of what we mean by predictive and descriptive parameters is given in Table 1 It is a rigged-up snippet of the user data where *PercCorr1* is the mean success rate - mean of *Correct* - over prior user trials including the current one. *PercCorr2* is the mean success rate over trials strictly prior to the current one. *PercCorr3* is the percent correct over all user trials.

PercCorr2 is an example of strictly predictive coding of the data, since at user trial t no information about performance of trial t is directly or indirectly incorporated into it. A model that would estimate a parameter for *PercCorr2* would capture the predictive nature of this value. *PercCorr1* and *PercCorr3* are

Table 1. Predictive vs. descriptive parameters (rigged up example)

User ID	Trial No.	Correct	PercCorr1	PercCorr2	PercCorr3
u11	1	1	1.00	null	0.60
u11	2	0	0.50	1.00	0.60
u11	3	0	0.33	0.50	0.60
u11	4	1	0.50	0.33	0.60
u11	5	1	0.60	0.50	0.60

the examples of descriptive coding of the data. *PercCorr1* incorporates the user performance and the current trial t and *PercCorr3* aggregates user performance over all trials: past, current, and future. Although *PercCorr1* and *PercCorr2* look much the same, models built using one or the other can differ greatly.

Clearly, predictive coding of the data is only possible when repeated measures are made. If each user contributes just one data point, only descriptive parameters can be constructed. There is no universal recipe for deciding when to include predictive or descriptive parameters into the model. From our experience, models that are built from repeated measures data (arguably, most of the user models are) and include both predictive and descriptive parameters are more stable and less prone to over-fitting than those that only include descriptive parameters.

3 Data

The dataset that we will use in this paper contains student activity recorded by a modified Bridge to Algebra (BTA) tutor by Carnegie Learning[1]. It was collected in several sixth and seventh grade classes at Pinecrest Academy Charter Middle School and covers 10 warmup sessions added to the main BTA curriculum of 61 existing BTA sections. Warmup sessions addressed the same topic as the forthcoming BTA tutor section. In each of the warmups, users (11-13 year old kids) were presented with 16 simple unscaffolded math problems randomly drawn from a pool of 24. Subjects were distributed across several experimental conditions differing in what accompanied problems 5 through 12 (worked problem, hint, or nothing at all). Subjects in a special *inference* condition were only given 8 problems.

For our modeling we used a subset of the data: the first warmup session addressing least common multiples. This data is comprised of 3616 problem trials (fill-in-the-blank exercises, worked problems, and hints were excluded) belonging to 255 students that completed all 16 assigned problems (8 in case of *inference* condition). Texts of two of the problems are given below as examples.

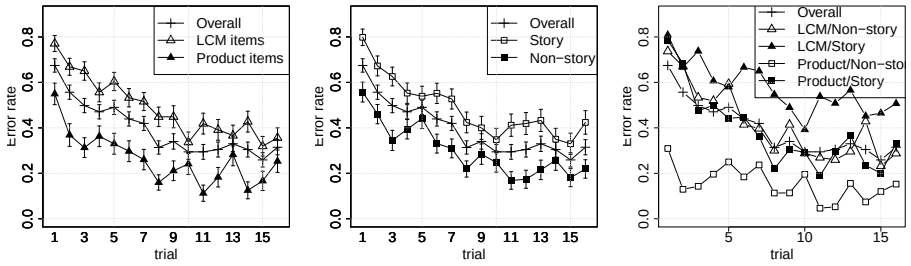
> *Problem example 1. Sally visits her grandfather every 4 days and Molly visits him every 6 days. If they are visiting him together today, in how many days will they visit together again?*

[1] http://www.carnegielearning.com/secondary-curricula/bta/

Problem example 2. What is the least common multiple 4 and 9?

The problem examples above have two important properties. First, problem 1 is a so-called *story* problem and problem 2 is a *non-story* problem. Story problems require additional abstraction or a use of a concrete strategy. In the literature, there could be found conflicting evidence on whether *story* problems were more difficult or not (see, for example [6, 5]). In our case, story problems are generally harder: overall mean success rate for *story* problems is 0.50 which is lower than the overall mean success rate for *non-story* problems that is 0.69. Out of 24 problems in the first warmup pool, 12 were *story* problems and 12 were *non-story* problems.

A second and, arguably, more important property of the problems is that in some cases the least common multiple (LCM) could be correctly obtained by multiplying the two inputs. In this case, the problem can be solved by applying *partial* problem-solving strategy. However, not all LCM's are equal to the product of the inputs. Problem example 2 is such problem, where LCM of 4 and 9 is $36 = 4 \times 9$. Problem example 1 is a case that requires *full* problem-solving strategy and a product of the inputs would give an erroneous result. Here, LCM of 4 and 6 is $12 \neq 4 \times 6$. In the problem pool, 10 were the problems that could be solve by multiplying the inputs. We will be calling them *Product* problems. In our data set, 14 problems were the ones, for which product of the inputs would yield and incorrect result. We will be referring to them as *LCM* problems. Naturally, *LCM* problems were harder and had 0.50 mean overall success rate, as compared to *Product* problems with 0.73 mean overall success rate.



(a) (b) (c)

Fig. 1. Error rates comparisons. Serifs in (a) and (b) depict 95% confidence intervals

Fig. 1 shows comparisons of error rate curves (compliment of the learning curve). As we can see in Fig. 1(a) and Fig. 1(b), *LCM* problems' and *story* problems' error rate curves respectively are reliably higher. When these two properties are crossed and four error rate curves are produced (Fig. 1(c)), *LCM/story* problems represent the most hard combination of the properties and *Product/non-story* - the least hard. Respective error rate curves are the highest and the lowest

on the graph. The other two – *LCM/non-story* and *Product/story* – are close to the overall curve.

4 Subject Parameters as Fixed vs. Random Effects

Mixed effects modeling approaches are now commonly used in many areas of science. Among other things, these approaches prescribe treating participant-specific model parameters as random effects [1]. A random effect is an effect that is sampled from a population to which statistical inferences are to generalize. Subjects are treated as random factors, because the goal of modeling is to capture effects pertaining not only to the individuals participating in a particular experiment, but to the subject population in general. Entering users as fixed factors (referred to as *fixed fallacy* in [3]), due to sampling variability, could make the model less generalizable and results would not transfer to similar datasets.

This argument may also apply to problem items as well that are often prescribed to be entered as random effects crossed with users (see, for example, [1]). In our dataset, however, problem items were not randomly drawn from a larger problem pool. Problem set was fixed by experimenters. Using random effects for problem items may refine the model further, based on the same principals that led us to use random effects for users. However, we leave determining a possible benefit of it for the further work.

In this section we are going to demonstrate the value of entering user proficiency parameters as random factors. We will do that on the basis of the Performance Factors Analysis (PFA) [9]. PFA is an educational data mining model. It was developed as an elaboration of the Additive Factors Model (AFM) [2] that in its turn is an extension of the Rasch item response model [7].

4.1 PFA Models

The PFA model uses the numbers of prior correct and incorrect responses as indicators of the strength of the student's mastery of a knowledge component (KC). Inclusion of the number of correct responses, in addition to capturing learning, allows PFA to track strength of students knowledge: the more correct responses are produced, the more it is likely that student's knowledge is already high. The number of incorrect plays the role of reflecting learning from errors and also acts as counter-balance, since errors are indicative of the relative weakness of the student's knowledge. Together, both corrects and incorrects not only make model sensitive to the quantity of each, but also the ratio of one to the other.

PFA's standard multiple logistic regression form is shown in Equation (1). Here, Pr denotes probability, $inv.logit$ is an inverse logistic function: $inv.logit(x) = 1/(1+e^{-x})$, Y_{ij} denotes the response of student $i \in [1, U]$ on $\mathrm{KC}_j \in [1, K]$, θ_i - coefficient of proficiency of user i, β_j - coefficient of difficulty of KC_j, γ_j - coefficient for the number of correct responses of the KC_j (success rate parameter), ρ_j - coefficient for the number of incorrect responses of the KCj (failure rate parameter), s_{ij} - user i success rate with KC_j, f_{ij} - user i failure rate with KC_j.

$$p_{ij} = \Pr(Y_{ij} = 1|\theta, \beta, \gamma, \rho) = inv.logit[\ (\theta_i + \sum_j (\beta_j + \gamma_j s_{ij} + \rho_j f_{ij})\ ] \quad (1)$$

θ and β parameters in PFA are always estimated in a descriptive fashion, since they capture overall KC difficulties and overall user proficiencies. Number of correct (s) and incorrect KC attempts (f) could be computed either descriptively or predictively (thus defining how success (γ) and failure (ρ) rate parameters are estimated). In this paper, we always compute attempt counts and respective success/failure rate parameters as predictive (like PercCorrect2 in Table 1).

Based on the PFA model shown in Equation (1), we build several variants. First one is the PFA without coefficient of user's proficiency (parameter θ_i, present in standard PFA is excluded). We will refer to it as PFA *ns* (no subject). This model is shown in Equation (2). Although this model disregards variability in user proficiency entirely, it is still potentially able to offer useful insights based on KC difficulties and success/failure rate parameters alone. Another variant of PFA, treats user proficiency parameters θ_i as random effects. Instead of estimating user proficiencies directly, it estimates their respective variance. In all other aspects this model is identical to the standard PFA model in Equation (1).

$$p_{ij} = \Pr(Y_{ij} = 1|\beta, \gamma, \rho) = inv.logit[\ \sum_j (\beta_j + \gamma_j s_{ij} + \rho_j f_{ij})\ ] \quad (2)$$

In our dataset problems (denoted by j subscript in the PFA model) initially were not indexed with KC's like in the original work on PFA [9]. We are going to use problem-solving strategies instead of KC's and will call them [problem] itemtypes, as was done in a later version of the PFA model [8]. Thus, we have two different itemtypes: *Product* - for the problems, where using a partial strategy is permissible, and *LCM* - where only the use of the full strategy would produce a correct result.

4.2 Model Comparison Results

Table 2 presents a summary of several fit statistics for the three PFA models. Namely, number of parameters (Par.) log-likelihood (LL), Bayesian Information Criterion (BIC), correlation of actual and expected accuracy across students (r_{AE}), area under ROC curve (A′) and sum of squared residuals (SSR). Judging just from these values, PFA seems to have an edge: LL and SSR are the lowest, r_{AE} and A′ are the highest. BIC, however, is the highest of all three models. PFA *ns* is the least successful and PFA *re* is roughly between the other two models. However, as our main thesis of the paper suggests, the *surface* statistics in Table 2 are not enough.

Table 3 is a summary of actual model parameters. Across all three models, *Product* itemtypes are consistently harder than *LCM* itemtypes: β_P intercepts are higher than β_L intercepts. In PFA *ns* model, success rate parameters γ_P and γ_L are both reliably above zero reflecting that users do learn from correct

responses as expected, more from corrects on *LCM* itemtypes. Errors on *Product* itemtypes *hurt* student performance ($\rho_P \leq 0$, p-value=0.000), while errors on *LCM* itemtypes (ρ_L), do not have a significant effect.

Table 2. Fit statistics of PFA models

Model	Par.	LL	BIC	r_{AE}	A′	SSR
PFA *ns*	6	-2133	4323	0.860	0.739	728.527
PFA	261	-1768	5674	1.000[2]	0.836	581.637
PFA *re*	7	-2123	4296	0.984	0.800	646.908

Table 3. Parameters of PFA models. Subscripts $_P$ and $_L$ refer to *Product* and *LCM* itemtypes respectively

	PFA *ns*			PFA			PFA *re*		
	Par.	Std.Err.	p-value[3]	Par.	Std.Err.	p-value	Par.	Std.Err.	p-value
β_P	0.452	0.077	0.000***	-0.569	0.098	0.010**	0.386	0.082	0.000***
β_L	-0.647	0.073	0.000***	-1.989	0.770	0.000***	-0.800	0.089	0.000***
γ_P	0.118	0.026	0.000***	-0.179	0.046	0.000***	0.046	0.033	0.162
ρ_P	-0.110	0.037	0.003**	0.716	0.082	0.000***	0.075	0.050	0.134
γ_L	0.354	0.026	0.000***	0.018	0.041	0.660	0.274	0.032	0.000***
ρ_L	-0.028	0.021	0.189	0.367	0.043	0.000***	0.081	0.028	0.004**
θ_i	N/A	N/A		-0.003	3.381		-0.008	0.731	
	Mean	Std.Dev.		Mean	Std.Dev.		Mean	Std.Dev.	

While PFA *ns* seems to be generally acceptable, there is one thing that raises caution. Namely, the model fits failure rate parameters to have negative or no effect on students' future performance and, while it seems plausible to expect at least a hint at learning from errors (cf. [4]). Instead, users actually get worse after failing the *Product* problems. Our explanation for it is that ρ_P in PFA *ns* model compensates for the absence of user proficiency parameters. The only way for PFA *ns* to distinguish higher achieving students (with fewer errors) from lower achieving students (with more errors) is to resort to error tracking. As a result, ρ_P is reliably negative. Because of that, PFA *ns* is not complex enough.

The PFA model presents quite a radical picture. Both failure rate parameters (ρ_P and ρ_L) are positive and very high, success rate parameter for *LCM* (γ_L) is indistinguishable from zero, while success rate parameter for *Product* (γ_P) is reliably negative. In addition, standard deviation of the user proficiency coefficient θ_i is dubiously high. Our intuition is that such parameter value *reversal* originates from optimizing the early performance using the fixed subject proficiency factors. A user will tend to perform at this fixed base performance

[2] The actual value is smaller than 1.000 and is equal to 0.999999999999998668

[3] Significance codes are: . – p≤0.1, * – p≤0.05, ** – p≤0.01, *** – p≤0.001

level, which, if already high, will need little change across practice (hence a low γ parameter). In contrast, if the fixed base is low, learning must still occur to capture the general increase in performance in the data. Since correct results are infrequent with low initial strength, the learning is forced to be captured by the ρ parameters.

The PFA *re* model is, arguably, the most accurate of the three and we argue that this is mainly due to the fact that user proficiencies are entered as random factors. Success/failure rate parameters for *LCM* itemtype are reliably greater than zero. Failure rate parameter ρ_L is almost four times smaller than success rate parameter γ_L. Nevertheless, the model detects some learning from mistakes as well. Success and failure rate parameters for *Product* itemtype are indistinguishable from zero now. A possible explanation for this could be that the model is not complex enough and results reported in Section 5.2 support this hypothesis. Variability of the user proficiency parameters looks reasonably constrained. At this point there seems to be no indication of problems with parsimony.

Table 4. Cross-validation of PFA models

Model	Data	mean(LL)	mean(BIC)	r_{AE}	A′	MSE
PFA *ns*	train	-0.590	1.196	0.859	0.739	0.201
	test	-0.594	1.243	0.852	0.735	0.203
PFA	train	-0.489	1.553	1.000	0.836	0.161
	test	-0.518	2.940	-0.723	0.544	0.308
PFA *re*	train	-0.587	1.193	0.984	0.800	0.179
	test	-0.585	1.234	0.567	0.716	0.209

To investigate the source of PFA's radical parameter values, we performed a cross-validation of the three PFA models discussed in this section. We performed 20 independent runs during which a 5-fold cross-validation was performed. Folds were stratified by users: 80% of the users were randomly chosen for training, 20% of the users were retained for testing. During each of the 20 runs, only one random split was performed. Model fit statistics reported in Table 4 are averaged across these 20 training and testing runs. When computing statistics for PFA and PFA*re* models, user proficiencies were set to zero value (mean of user proficiencies in Table 3) for the test dataset, since users in test dataset were not seen by these models before.

As we can see from Table 4, PFA no longer has the edge over PFA *ns* and PFA *re*. Despite, mean log-likelihoods having smaller absolute values for PFA model, the rest of the metrics put it at disadvantage. Mean BIC for test dataset as compared to training dataset goes up only slightly for PFA *ns* and PFA *re*, while for PFA it almost doubles. r_{AE} of PFA model in train dataset drops radically from 1.000 to -0.723 while remaining hight and positive for PFA *ns* and PFA *re*. A′ for PFA's test dataset drops almost to the random-guessing baseline level of 0.500, while changes from train to test dataset do not shrink A′ for PFA

ns and PFA *re* models considerably. Mean squared error for PFA doubles on the test and only goes up a little for PFA *ns* and PFA *re*.

As a result, PFA model with fixed-factor user proficiencies seems to be terribly over-fit. At the same time PFA *ns* and PFA *re* hold quite well. A relatively worse behavior of PFA *re* revealed in steeper drops of A′, MSE, and especially r_{AE} between train and test sets, can be attributed to the fact that in training set user proficiencies are effectively removed (set to zero). User proficiency agnostic PFA *ns* performs on the test set performs slightly better. Despite this, PFA *ns* is our preferred model, for we think that its abilities to reflect learning from errors and account for variability in user proficiencies are very important.

5 Model Complexity and Parsimony

The results of fitting PFA models from the previous sections show that there is room for improvement, at least in terms of complexity. In this section we are going to suggest two extensions to the PFA *re* model and discuss resulting changes with respect to complexity and parsimony. Building on the results of the previous section, we will only fit models with user proficiencies entered as random factors.

5.1 Extended PFA Models

Our first extension to the PFA *re* model addresses the definition of itemtypes. In Section 4.2 above, we specified itemtypes according to problem-solving strategies: *Product* and *LCM*. However, another property of the problems - whether it is a story problem or not - was disregarded. As it is known from literature, students might react to story problems differently (cf. [6, 5]). Incorporating information of whether a problem is a story problem into the model could potentially benefit it and help reflect the problem semantics more comprehensively. This extended PFA, that we will refer to as *ext* PFA1 *re* is virtually identical to the PFA *re*. The difference is that the itemtypes are now four: *Product story*, *Product non-story*, *LCM story*, and *LCM non-story*.

In a second extended model we are going to use four problem itemtypes types again. In addition, a new term that captures running percent correct on all prior problem itemtypes will be entered. Current attempt will be excluded and on the first attempt the value of the percent correct would be set to 0.5. This model will be referred to as *ext* PFA2 *re* and is shown in Equation (3). There, c_i denotes user's percent correct on prior problem attempts, and δ is the model coefficient for it.

$$p_{ij} = \Pr(Y_{ij} = 1|\theta, \beta, \gamma, \rho, \delta) = inv.logit[\ \theta_i + \delta c_i + \sum_j (\beta_j + \gamma_j s_{ij} + \rho_j f_{ij})\ ] \quad (3)$$

5.2 Model Comparison Results

As we can see from Table 5, both extended models have an edge over PFA in terms of log-likelihood, BIC, A′, and SSR statistics. When compared to each other, extended models are hardly distinguishable from each other, although, according to χ^2 test, *ext* PFA2 *re* has an edge (χ^2=5.394, p-value=0.020) . Let us, however, turn to Table 6 and compare model parameters.

Table 5. Fit statistics of extended PFA models

Model	Par.	LL	BIC	r_{AE}	A′	SSR
PFA *re*	7	-2123	4296	0.984	0.800	646.908
ext PFA1 *re*	13	-2030	4149	0.983	0.826	604.700
ext PFA2 *re*	14	-2016	4152	0.967	0.812	626.857

Table 6. Parameters of extended PFA models. Subscripts $_P$, $_L$, $_S$, and $_{nS}$ refer to *Product*, *LCM*, *story*, and *non-story* itemtypes and their combinations respectively

PFA *re*				*ext* PFA1 *re*				*ext* PFA2 *re*		
	Par.	Std.Err.	p-value		Par.	Std.Err.	p-value	Par.	Std.Err.	p-value
β_P	0.386	0.082	0.000^{***}	β_{PnS}	0.945	0.086	0.000^{***}	0.687	0.085	0.000^{***}
				β_{PS}	-0.140	0.085	0.000^{***}	-0.382	0.084	0.000^{***}
β_L	-0.800	0.089	0.000^{***}	β_{LnS}	-0.335	0.099	0.000^{***}	-0.572	0.142	0.000^{***}
				β_{LS}	-1.419	0.084	0.000^{***}	-1.642	0.083	0.000^{***}
γ_P	0.046	0.033	0.162	γ_{PnS}	0.004	0.052	0.943	-0.032	0.054	0.549
				γ_{PS}	0.142	0.060	0.017^{*}	0.125	0.059	0.032^{*}
ρ_P	0.075	0.050	0.134	ρ_{PnS}	0.071	0.105	0.500	0.045	0.100	0.656
				ρ_{PS}	0.086	0.065	0.186	0.085	0.053	0.181
γ_L	0.274	0.032	0.000^{***}	γ_{LnS}	0.347	0.050	0.000^{***}	0.321	0.050	0.000^{***}
				γ_{LS}	0.226	0.061	0.000^{***}	0.226	0.060	0.000^{***}
ρ_L	0.081	0.028	0.004^{**}	ρ_{LnS}	0.251	0.052	0.000^{***}	0.244	0.051	0.000^{***}
				ρ_{LS}	-0.058	0.047	0.220	-0.034	0.047	0.461
				δ_i	N/A	N/A	N/A	0.606	0.240	0.012^{*}
θ_i	-0.008	0.547			-0.008	0.780		-0.006	0.660	
		Mean	Std.Dev.		Mean	Std.Dev.		Mean	Std.Dev.	

In Table 6 we see that β itemtype complexity intercepts in extended models are lower for *LCM* itemtype than for a corresponding *Product* itemtype, just like in PFA models (see Table 3). However, *story* property adds additional differentiation. Within *LCM/Product* levels, *story* intercepts are always lower reflecting the fact that, in our dataset, story problems are harder. This phenomenon can be traced to some other pairs of *story* and *non-story* significant parameters

(e.g. $\gamma_{LS} < \gamma_{LnS}$ in *ext* PFA1 *re* model). In addition, in both extended models *LCM/non-story* has lower intercept than *Product/story*.

The first extended PFA model (*ext* PFA1 *re*) provides and interesting specification of the PFA model (PFA *re*). The γ_P – success rate parameter for *Product* itemtypes – had no significant effect in PFA. When split in the first extended model, the *story* part of it (γ_{PS}) is now significant. Namely, successes on *Product/story* problems are indicative of student performance. Failure rate parameters for *Product* remain having no detectable effect.

Success rate parameters for *LCM* itemtypes remain positively predictive of student successes. Success rate parameter for *LCM/story* being smaller than for *LCM/non-story*, suggesting that for the *LCM/story* itemtypes being the hardest complexity inhibits the benefit of correct responses. This phenomenon could also be seen in failure rate parameters. This is too reflected in that errors for *LCM/non-story* itemtypes (ρ_{LnS}) have positive effect on learning, while errors for *LCM/story* itemtypes (ρ_{LS}) have no statistically detectable influence. Overall we can conclude that the complexity that the first extended PFA model adds to the PFA model not only improves the fit, but also facilitates better understanding of student learning and problem domain properties.

The second extended PFA model (*ext* PFA2 *re*) that has one additional parameter – users overall problem percent correct – is in our case an example of lack of parsimony. Although the new parameter is a significant predictor of student performance, it does not bring any additional insights into better understanding of student learning, since it is highly correlated with the individual learning rates that come from the same data. All it does is reduce variance of most of the model parameters from first extended model (including random-effect user proficiencies) while significance levels of the model parameters mostly stay the same. Thus, the value of increased complexity of the model *ext* PFA2 *re* as compared to the model *ext* PFA1 *re* is questionable at best.

6 Conclusions

The motto of the statistical modeling, repeated by scores of instructors, is that every model should be checked against a preconceived theory rather than judged solely by model fit statistics. In this paper we tried to trace that statement to a list of possible pitfalls that user modelers could find themselves in if they do the opposite. The list of potential problems is, likely, incomplete, but the ones we mentioned, if avoided, would arguably make researchers' life a lot easier. Namely, models would be more meaningful and interpretable, would generalize better, and would be less prone to over-fitting.

Throughout the paper, we used a user modeling approach called Performance Factors Analysis (PFA) as an method. The advantages and drawbacks of PFA and its variations that we constructed are most likely specific to PFA only. Should other user modeling methods be used, the magnitude of the effects we discussed or the effects themselves could change. Similarly, the nature of outcomes could change too if a different dataset is used, with richer problem attributes' seman-

tics, for example. However, because of the generality of the issues we addressed, following the advice in this paper is likely benefit most efforts to understand data using mathematical modeling.

Acknowledgments This research was supported by the U.S. Department of Education (IES-NCSER award #R305B070487) and was also made possible with the assistance and funding of Carnegie Learning Inc., the Pittsburgh Science of Learning Center, DataShop team (NSF-SBE award #0354420) and Ronald Zdrojkowski.

References

1. Baayen, R. H., Davidson, D. J., & Bates D. M. (2008) Mixed-effects modeling with crossed random effects for subjects and items. Journal of Memory and Language, 59(4), 390-412.
2. Cen, H., Koedinger, K. R., & Junker, B. (2008) Comparing Two IRT Models for Conjunctive Skills. In Woolf, B.P, Aïmeur, E., Nkambou, R., and Lajoie, S. (Eds.), Proceedings of the 9th international conference on Intelligent Tutoring Systems (ITS '08), Springer-Verlag, Berlin/Heidelberg, (pp. 796-798).
3. Clark, H. H. (1973). The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. Journal of Verbal Learning and Verbal Behavior, 12, 335-359.
4. Corbett, A. T., & Anderson, J. R. (2001). Locus of feedback control in computer-based tutoring: impact on learning rate, achievement and attitudes. Proceedings of CHI 2002, Human Factors in Computing Systems (March 31April 5, 2001, Seattle, WA, USA), ACM, pp. 245-252.
5. Cummins, D. D., Kintsch, W., Reusser, K., & Weimer, R. (1988). The role of understanding in solving algebra word problems. Cognitive Psychology, 20, 405-438.
6. Koedinger, K. R., & Nathan, M. J. (2004). The real story behind story problems: Effects of representation on quantitative reasoning. Journal of the Learning Sciences, 13, 129-164.
7. van der Linden, W. J., Hambleton, R. K. (eds.): Handbook of Modern Item Response Theory. Springer Verlag, New York, NY (1997).
8. Pavlik, P. I., Cen, H., & Koedinger, K. R. (2009). Learning factors transfer analysis: Using learning curve analysis to automatically generate domain models. In Barnes, T., Desmarais, M., Romero, C. & Ventura, S. (Eds.), Proceedings of The 2nd International Conference on Educational Data Mining, Cordoba, Spain (pp. 121-130).
9. Pavlik Jr., P. I., Cen, H., & Koedinger, K. R. (2009). Performance factors analysis – A new alternative to knowledge tracing. In V. Dimitrova & R. Mizoguchi (Eds.), Proceedings of the 14th International Conference on Artificial Intelligence in Education. Brighton, England.
10. Pitt, M. A., Myung, I. J., & Zhang, S. (2002) Toward a method of selecting among computational models of cognition. Psychological Review, 109(3), 472-491.