Performance Factors Analysis – A New Alternative to Knowledge Tracing

Philip I. PAVLIK JR [a,1], Hao CEN [b], Kenneth R. KOEDINGER [a]
[a] *Human Computer Interaction Institute, Carnegie Mellon University, USA*
[b] *Machine Learning Department, Carnegie Mellon University, USA*

Abstract. Knowledge tracing (KT)[1] has been used in various forms for adaptive computerized instruction for more than 40 years. However, despite its long history of application, it is difficult to use in domain model search procedures, has not been used to capture learning where multiple skills are needed to perform a single action, and has not been used to compute latencies of actions. On the other hand, existing models used for educational data mining (e.g. Learning Factors Analysis (LFA)[2]) and model search do not tend to allow the creation of a "model overlay" that traces predictions for individual students with individual skills so as to allow the adaptive instruction to automatically remediate performance. Because these limitations make the transition from model search to model application in adaptive instruction more difficult, this paper describes our work to modify an existing data mining model so that it can also be used to select practice adaptively. We compare this new adaptive data mining model (PFA, Performance Factors Analysis) with two versions of LFA and then compare PFA with standard KT.

Keywords. knowledge tracing, adaptive modeling, educational data mining

Introduction

Adaptive instruction systems often utilize a computational model of the effect of practice on KCs (i.e. knowledge components, which may include skills, concepts or facts) as a means to individually monitor and adapt to student learning. Similarly, efforts to analyze and determine the KCs in existing data sets (educational data mining) use computational models of the effect of practice on KCs as a means to understand a learning domain with the hope that results can be applied to improving existing systems from which the data was drawn. Although these applications are similar, often different models are used for each task. Because of this it is often not easy to determine how one moves from an educational data mining result directly to an adaptive tutoring system. In an attempt to address this problem, this paper describes how we have taken a model that is more applicable to educational data mining and modified it so that it retains its advantages for educational data mining while gaining the ability to make the individual student KC inferences that can be used in an adaptive manner.

Our ultimate goal is to foster what can be called "closed loop" tutor development systems. Such systems can be referred to as closed loop because they would require very little technical expertise to add items to the system, gather data for those items, fit a model to the entire learning system, and then decide which items the model suggests

[1] Corresponding Author.

are poorly learned or do not transfer to other content and drop those items. This development loop begins each cycle with a portion of new items being added, and ends each cycle with a refined system that retains what is useful and discards what is useless. Such a system would allow educators to take a much larger role in tutor development and improve the speed of tutor development. In contrast to such a system, currently items are added somewhat arbitrarily to tutors, parameters may be fixed across years of use of a tutoring system, models are rarely fit to existing tutor data, and when models are fit it is rare that the results transfer easily to an adaptive system in the classroom.

The educational data mining model we begin with is the Learning Factors Analysis (LFA) model. This model has been used as a data mining tool in various ways. For example, as part of a search algorithm this model has been used to split knowledge components along multiple factors to determine the KC to item assignment (also known as the Q matrix) that represents the data best while still controlling for model complexity [2]. The LFA model captures three important factors influencing the learning and performance of KCs. First, it captures subject ability in a single parameter for each subject. Second, it captures KC easiness with a single parameter for each KC. Third, it captures the learning rate for each KC with a single parameter for each KC. However, while the LFA model has considerable power to fit data given this formulation, it has very little power to dynamically differentiate between individual KCs for particular students because ignores the correct and incorrect response produced by the student. Therefore, while this model is sensitive to practice frequency, it offers very poor affordances for adapting this frequency because it ignores the evidence of learning in the correct and incorrect responses produced by each student. Essentially, the LFA model says that all students accumulate learning in an identical fashion. Of course, this is not true, and therefore the LFA model is unsuitable for adaptive learning algorithms. (This issue also causes problems for datamining since it means that student level KC covariance can only be modeled as a function of performance frequency.)

As a contrast, it helps to consider the knowledge tracing (KT) procedure pioneered by Atkinson[3] and developed by Corbett[1]. In KT (Corbett's version) there are 4 parameters fit to each KC which represent initial learning, learning rate, guess parameter, and slip parameter. One advantage of this model is that these 4 parameters are interpretable, so it is easy to understand their effects on performance in the model. These 4 parameters can be fit from prior student data for each KC in a domain, and they allow a tutoring program to use a student's prior history of performance with items for a KC to feed into the model equation so as to update the current estimate of student learning based on the students performance. Having this dynamic estimate provides a powerful ability to track individual differences with each KC. This personalized "model overlay" can then be used to make personalized decisions about which KCs a student has learned, and which KCs need more practice. Because of its simplicity and accuracy, the KT model has been used extensively in commercial tutors in addition to many experimental studies. All 3 of the Carnegie Learning Inc. datasets described further in the paper were from tutors that used KT.

While KT has desirable properties, the LFA model has some advantages that make it more tractable for data mining and performance optimization applications. First, the LFA model has been more extensively investigated as a solution to the problem of multiple KC performances[4,5]. A solution to this issue is useful because often tutor designers create practice steps where the student's response requires multiple KCs. If our model cannot explicitly capture these multiple KC performances, it cannot be used with datamining procedures to search for the optimal structure for such a conjunctive

KC performance model. Although the KT model has been explored for such purposes by multiplying the probabilities from the model when multiple KCs occur [6], such models have not been used to search for domain models as with LFA. The LFA model in this paper models conjunction by summing the contributions from all KCs needed in a performance. This sort of "compensatory" model of multi-KC behaviors allows lack of one KC to compensate for the presence of another in addition to showing conjunctive effects.

Second, models such as LFA (which produce a real valued estimate of strength for each KC) have been frequently employed predict the duration of each action (latency in the cognitive psychology literature). KT models might be explored in an attempt to build such a mechanism, but such work would be novel. Therefore, LFA seems more appropriate for advanced adaptive tutoring since LFA allows us to develop practice selection algorithms that maximize the learning rate (normally computed as learning gain per unit of time)[7] as an alternative to current algorithms which simply schedule practice until 95% mastery of the KC. Further model complexity than we describe in this paper is necessary to implement this goal, but predicting latency is a first step and may be crucial to using an adaptive practice algorithm effectively [3].

Because of these possible advantages to LFA and our desire to speed tutor development by closing the development loop, it would be desirable if we could formulate a version of LFA that could be used adaptively. To do this the following sections explain how we have reconfigured LFA to create a version we call Performance Factors Analysis (PFA). PFA provides the adaptive flexibility to create the model overlay we need for it to be used adaptively in a tutor, while retaining the data mining advantages that will allow us to use it in model search procedures.

1. Performance Factors Analysis

LFA's standard form is shown in Equation 1, where *m* is a logit value representing the accumulated learning for student *i* (ability captured by α parameter) using one or more KCs *j*. The easiness of these KCs is captured by the β parameters for each KC, and the benefit of frequency of prior practice for each KC is a function of the *n* of prior observations for student *i* with KC *j* (captured by the addition of γ for each observation). Equation 2 is the logistic function used to convert *m* strength values to predictions of observed probability. This model is an elaboration of the Rasch item response model which has an equivalent form to Equation 1 with γ set to 0 and only a single β value.

$$\mathrm{m}(i, j \in KCs, n) = \alpha_i + \sum_{j \in KCs} (\beta_j + \gamma_j n_{i,j}) \tag{1}$$

$$\mathrm{p}(m) = \frac{1}{1 + e^{-m}} \tag{2}$$

Because of its usefulness in various data mining applications we might wish to reconfigure this model so that it could be used to engineer practice selection in a similar way as is done using the KT model. One way to reconfigure the model is to make it sensitive to the strongest indicator of student learning: performance.

Performance is indicative of student learning for two reasons[8]. First, correct responses are strongly indicative that current strength is already high, therefore, given a correct response it will help our model to increase our strength estimate. Second, correct responses may simply lead to more learning than incorrect responses. This may be due to intrinsically greater processing during the production of a correct response, or perhaps due to ineffective review procedures after incorrect responses. However, while making the model sensitive to correctness is a good start, it also seems useful to make the model specifically sensitive to incorrectness. Sensitivity to incorrectness allows incorrectness to act as indicator and measure of learning in an inverse to correctness. Together, the inclusion of both correctness and incorrectness in the model will make it sensitive to not only the quantity of each, but also the relative ratio of correct to incorrect. See that in Equation 3, α has been removed from the model since it is not usually be estimated ahead of time in adaptive situations (however, as noted by Corbett, models that do track subject level learning variability can greatly improve model adequacy[1]). β has been previously explained, *s* tracks the prior successes for the KC for the student, *f* tracks the prior failures for the KC for the student, and γ and ρ scale the effect of these observation counts. Equation 2 is still applied for conversion to probability predictions. (Again, the model can be used in a compensatory fashion for observations requiring multiple KCs by summing the βs and γ and ρ frequency components for all *j* KCs needed.) We call this model PFA (Performance Factors Analysis).

$$m(i, j \in KCs, s, f) = \sum_{j \in KCs} (\beta_j + \gamma_j s_{i,j} + \rho_j f_{i,j}) \tag{3}$$

We fit the parameters (β, γ, α and/or ρ) for each model (LFA or PFA) to maximize loglikelihood of the model for 4 datasets. We also included a simpler third model (LFA ns) that was equivalent to LFA, but without any student parameter. The comparison of these three models allows us to see the improvement the PFA model over the LFA ns model which is identical but for the new performance accounting. We expect this comparison to set a lower bar on acceptability. In contrast, while full LFA has a powerful ability to capture individual differences because of the α parameter, by comparing PFA to it we can see how well our adaptive method approaches the accuracy of standard non-adaptive LFA.

These 4 datasets were from various sources. Fractions and Algebra (grades 5-8) were subsections of units from the Bridge to Algebra Cognitive Tutor from Carnegie Learning. The Geometry data (grades 9-12) were from the Angles and Areas units of the Carnegie Learning Geometry Cognitive Tutor. The Physics dataset (grades 9-12) comes from the Andes Physics tutoring system courtesy of VanLehn. Both geometry and physics datasets were download directly from the Pittsburgh Science of Learning Center DataShop service. All of these datasets came with preset domain content KC labels for each performance. These labels designated what KC was hypothetically responsible for the performance. Table 1 shows the organization of the data files for a short example sequence for a KC for a student (with fit PFA model values shown), which shows how each response has associated student, KC, subgoal and correctness values, from which the model predictions were computed. When subgoal (and student and response) maintained the same value for consecutive rows, this indicated that multiple KCs were coded for a single response at that subgoal.

Table 1. Example of data and model format.

Student	Correct	Subgoal	Skill	Model Pred.
a51864	1	a51864C101019	S44IdentifyGCFonenumbermultipleofother	0.6058313
a51864	0	a51864C10810	S44IdentifyGCFonenumbermultipleofother	0.6351781
a51864	1	a51864C101020	S44IdentifyGCFonenumbermultipleofother	0.6089495
a51864	1	a51864C10796	S44IdentifyGCFonenumbermultipleofother	0.6382028
a51864	1	a51864C101021	S44IdentifyGCFonenumbermultipleofother	0.6664663
…	…	…	…	…

1.1. Model Comparison Results

Table 2 shows the results of the comparison for several fit statistics, and also lists the number of parameters and observations. According to Bayesian Information Criterion (*BIC*), and loglikelihood (*LL*), LFA is marginally superior in 3 of 4 datasets. While this is to be expected considering that LFA includes a subject parameter unlike the other 2 models, we can also see that the new PFA version ties (*LL*) or beats (*BIC*) LFA in the Fractions dataset. Despite the fact that LFA is better, the PFA model compares well to LFA relative to the LFA no subject (ns) model. Although the BIC values suggest overfitting relative to the other models, the mean absolute deviation for a 7-fold crossvalidation (MAD CV) shows the model generalizes well. This demonstrates that the new mechanism is working to pick up individual differences nearly as effectively as a subject parameter. These comparisons demonstrate that the PFA model is a new alternative that may be useful for detecting and reacting to student learning in a tutor. The fact that the subject parameter in LFA captures slightly more student variability than the performance accounting in PFA implies that future work might further improve the adaptive PFA model by adaptively estimating a subject parameter.

Table 2. Comparison of the 3 LFA model versions for the 4 datasets.

Dataset	Model	Par.	Obs.	*LL*	*BIC*	*MAD CV*
Physics						
	LFA ns	376	4.093E+4	-2.124E+4	4.648E+4	0.349
	LFA	451	4.093E+4	-2.074E+4	4.627E+4	0.340
	PFA	564	4.093E+4	-2.099E+4	4.797E+4	0.346
Geometry						
	LFA ns	240	4.478E+4	-2.07E+4	4.398E+4	0.305
	LFA	274	4.478E+4	-2.011E+4	4.316E+4	0.295
	PFA	360	4.478E+4	-2.015E+4	4.416E+4	0.297
Algebra						
	LFA ns	276	4.657E+4	1.629E+4	3.554E+4	0.210
	LFA	387	4.657E+4	1.564E+4	3.545E+4	0.203
	PFA	414	4.657E+4	1.58E+4	3.605E+4	0.204
Fractions						
	LFA ns	128	1.01E+5	-3.689E+4	7.525E+4	0.216
	LFA	269	1.01E+5	-3.533E+4	7.377E+4	0.208
	PFA	192	1.01E+5	-3.533E+4	7.287E+4	0.207

2. Performance Factors Analysis compared to Knowledge Tracing

KT is based on a 2 state Markov model with 4 parameters controlling the probability of these 2 states, learned or unlearned. The 4 parameters, *L0*, *T*, *G*, and *S*, stand for initial learning probability, learning transition probability, guess probability and slip

probability. Full details of how these parameters are used to compute predictions for a series of practices has been explained previously[1]. To summarize, *L0* is the estimate of the learned state for the first practice, and *T* describes the probability of transition from unlearned to learned (the learning rate) after each practice. *G* and *S* are used to set floor and ceiling levels of performance, and make the inference from the students response history non-deterministic (e.g. if they get it right, it could have been a guess, and if they get it wrong, it could have been a slip.)

We were interested in comparing this model with the PFA model since if the PFA model should prove comparable or better, this would, given the advantages of PFA model mentioned in the introduction, constitute strong support for using the PFA more extensively in model-based adaptive instruction systems. To produce this comparison we used the 4 datasets used in the previous section. However, the previous comparisons (Table 2) applied the PFA model with the original multiple KCs per performance model that was coded in the datasets. We could not do this with the KT model since the KT model only allows one KC per step. To resolve this problem, we recoded the data by splitting each multiple KC performance into multiple single KC performances. (Of course, this increases the effective number of observations, but seems to do so in a way that favors neither model.) We then fit both the PFA and KT models to the datasets.

Unlike PFA, which takes the form of standard logistic regression and therefore allows fast optimization by computing the true solution gradient to maximize the loglikelihood, KT has less well explored options for finding parameters. We used preexisting unpublished code written by Beck and Leszczenski and adapted by Cen to find the KT 1 model for each dataset, and we used preexisting unpublished code written by Baker (which worked by performing an exhaustive search of the parameter space) to find the KT 2 model for each dataset. We used both of these options so that it would seem implausible that our method of fitting the parameters could be the cause of any advantage we found. Further, while PFA rarely produces near 0 or 1 predictions, occasionally KT can produce near these values which might inflate *LL* comparisons. To mitigate this issue, we used a prediction floor of 0.01 and a ceiling of 0.99 when computing the *LL* and *r* after the KT models had been fit according to the algorithms in the preexisting code. Additionally, PFA was bounded to a minimum of 0 γ to prevent over fitting from resulting in negative learning rates, and both KT versions were bounded to a maximum slip probability of 0.1 and a maximum guess probability of 0.3, values suggested by Corbett as appropriate to prevent over fitting. In keeping with the preexisting settings in the code, KT 2 was also bounded initial learning and transition probabilities at maximums of 0.85 and 0.3, respectively.

2.1. Model Comparison Results

Table 3 shows the results of the comparison for several fit statistics. While the differences are not large, the PFA model has better *LL*, *BIC*, *r* and *A'*. (*A'* has been described previously [9].) To get a better idea as to the meaning of the difference between the two models we did 2 subsidiary analyses. First, we supposed that the poor loglikelihoods for KT were caused by the fitting procedure. Specifically, while the KT 2 code used exhaustive bounded search, it attempted to minimize the sum of squared error (*SSE*) rather than maximize the sum loglikelihood. To see if this difference was responsible for the difference in fit for the models, we modified the KT 2 code to fit using the sum *LL* instead of *SSE* and tested the modified model (KT 2LL) for the

fractions and geometry datasets. For these two datasets, the result showed that using the *SSE* fit statistic did not appear to be driving misfit.

We also analyzed the learning curves produced by each model in the fractions and geometry datasets. Specifically, we looked at the predicted performance probability for each repeated observation with a KC conditional on the previous response for that KC. We noticed that the KT model had a tendency to predict much worse performance after a failure than did PFA, with the observed average in the data (for performance following failure) falling between the two model predictions, but much closer to PFA. Because of this pattern, one can speculate that the KT model has problems because it over estimates the importance of failure relative to correctness when attributing student learning. This effect is inherent in the KT model's assumption that a single failure (assuming a slip has not occurred) indicates the KC is unlearned. In contrast the PFA model assumptions mean a more gradual adjustment if a student fails a single trial. Although it could be argued that this is an effect of bounding the slip parameter, it also could be argued that the slip parameter becomes implausibly high in many cases (a problem with identifiability) when it is left unbounded.

Table 3. Comparison of the KT and PFA model versions for the 4 datasets.

Model	Data	Par.	Obs.	*LL*	*BIC*	*r*	*A'*
Physics							
	KT 1	752	4.099E+4	-2.234E+4	5.267E+04	0.326	0.708
	KT 2	752	4.099E+4	-2.25E+4	5.249E+04	0.326	0.705
	PFA	564	4.099E+4	-2.105E+4	4.809E+04	0.358	0.719
Geometry							
	KT 1	480	2.102E+5	-1.322E+5	2.703E+05	0.270	0.660
	KT 2	480	2.102E+5	-1.409E+5	2.877E+05	0.282	0.667
	KT 2LL	480	2.102E+5	-1.376E+5	2.811E+05	0.269	0.654
	PFA	360	2.102E+5	-1.223E+5	2.490E+05	0.305	0.685
Algebra							
	KT 1	552	1.37E+5	-5.867E+4	1.239E+05	0.224	0.687
	KT 2	552	1.37E+5	-5.795E+4	1.224E+05	0.247	0.692
	PFA	414	1.37E+5	-5.596E+4	1.168E+05	0.272	0.711
Fractions							
	KT 1	256	1.043E+5	-3.711E+4	7.718E+04	0.308	0.706
	KT 2	256	1.043E+5	-3.711E+4	7.718E+04	0.306	0.692
	KT 2LL	256	1.043E+5	-3.707E+4	7.710E+04	0.307	0.696
	PFA	192	1.043E+5	-3.643E+4	7.508E+04	0.320	0.723

3. Conclusion

PFA was described and compared to KT[1]. This comparison was highly relevant for the AIED audience because the "artificial intelligence" component in educational software often uses KT to create a student model overlay that allows the software to adapt to the student as learning progresses. Our results suggested that the PFA model was somewhat superior to the KT model overall. Secondary analysis suggested that the KT model assumption that a performance error indicates that (unless a slip occurred) the KC is unlearned was an exaggeration of the data. In contrast, PFA uses a parameter to scale how much is inferred by the model in the case of performance error, and this mechanism resulted in much less aggressive adjustment in prediction after a single error with a KC. This more gradual reaction to errors seemed to drive the advantage seen for the PFA model. Although KT has other advantages such as the fact that it results in different predictions depending on the order of practice, the misfit we saw

seems to depend on the KT model assumption that student knowledge is well represented as a Bernoulli distribution. Many results in the psychological literature, (e.g. forgetting results and overlearning results) suggest that the knowledge state might be better modeled by a continuous distribution that represents strength of learning, rather than a discrete probability distribution. By assuming that learning is continuous, the PFA model naturally lends itself to gradual adjustments in response to errors.

The accuracy advantage should also be considered in light of the two advantages described in the introduction. First, PFA can be applied to conjunctive and compensatory situations in a way that may be more difficult to accomplish with KT model. This advantage may allow greater complexity when using the model to search for how the domain should be split into KCs and how those KCs may combine in certain performances. Second, the PFA model produces the logit value which can be converted to a prediction of performance latency or duration. Having the ability to predict such action duration allows the model to be used in instructional engineering since it provides an estimate of the cost of each action. Knowing the cost of each action is an essential requisite in making decisions about the optimal action to take[3].

Acknowledgements

This research was supported by the U.S. Department of Education (IES-NCSER) #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) #0354420 and Ronald Zdrojkowski.

References

[1] Corbett, A.T., Anderson, J.R., Knowledge tracing: Modeling the acquisition of procedural knowledge, *User Modeling and User-Adapted Interaction* **4** (1995), 253-278.

[2] Cen, H., Koedinger, K.R., Junker, B., Learning Factors Analysis - A general method for cognitive model evaluation and improvement: *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*. Springer Berlin / Heidelberg (2006), 164-175.

[3] Atkinson, R.C., Ingredients for a theory of instruction, *American Psychologist* **27** (1972), 921-931.

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

[5] Cen, H., Koedinger, K.R., Junker, B., Comparing two IRT models for conjunctive skills, In: Woolf, B., Aimer, E., Nkambou, R. (eds.): *Proceedings of the Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, Montreal, Canada, (2008).

[6] Pardos, Z.A., Beck, J.E., Ruiz, C., Heffernan, N.T., The composition effect: Conjunctive or compensatory? An analysis of multi-skill math questions in ITS, In: Baker, R.S., Barnes, T., Beck, J.E. (eds.): *Proceedings of the 1st International Conference on Educational Data Mining*, Montreal, Canada, (2008), 147-156.

[7] Pavlik Jr., P.I., Anderson, J.R., Using a model to compute the optimal schedule of practice, *Journal of Experimental Psychology: Applied* **14** (2008), 101-117.

[8] Pavlik Jr., P.I., Understanding and applying the dynamics of test practice and study practice, *Instructional Science* **35** (2007), 407-441.

[9] Fogarty, J., Baker, R.S., Hudson, S.E., Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction: *Proceedings of Graphics Interface 2005*, Vol. 112. Canadian Human-Computer Communications Society, Victoria, British Columbia, (2005), 129-136.