Using autoKC and Interactions in Logistic Knowledge Tracing

Philip I. Pavlik Jr. and Liang Zhang
University of Memphis
ppavlik, lzhang13@memphis.edu

ABSTRACT

A longstanding goal of learner modeling and educational data mining is to improve the domain model of knowledge that is used to make inferences about learning and performance. In this report we present a tool for finding domain models that is built into an existing modeling framework, logistic knowledge tracing (LKT). LKT allows the flexible specification of learner models in logistic regression by allowing the modeler to select whatever features of the data are relevant to prediction. Each of these features (such as the count of prior opportunities) is a function computed for a component of data (such as a student or knowledge component). In this context, we have developed the "autoKC" component, which clusters knowledge components and allows the modeler to compute features for the clustered components. For an autoKC, the input component (initial KC or item assignment) is clustered prior to computing the feature and the feature is a function of that cluster. Another recent new function for LKT, which allows us to specify interactions between the logistic regression predictor terms, is combined with autoKC for this report. Interactions allow us to move beyond just assuming the cluster information has additive effects to allow us to model situations where a second factor of the data moderates a first factor.

Keywords

Domain models, learner modeling, logistic regression, knowledge tracing, adaptive learning

1. INTRODUCTION

Quantitative models of learning, used to predict performance and make pedagogical decisions, have a long history [1; 2]. To do this prediction effectively, models typically assign sets of problems or items specific skill tags (often called knowledge components, or KCs). Having such an identification of knowledge components allows a system to monitor mastery of skills. The matrices representing these item assignments to skills are called Q-matrices [4]. Because the act of tracing student learning is so important for pedagogy, the assignment of items to KCs is crucially important to understand for systems to make pedagogical choices. Without such an assignment, a system would conceivably need to schedule all practice items for practice to ensure mastery, since there would be no way to make inferences among the items belonging to a KC. In other words, a higher quality domain model should presumably result in better pedagogical decisions in a system due to improved ability to make inferences about other items based on sampling items for each KC. This paper is about improving these critical domain models, a tradition that has included much prior work [3; 5; 17; 19; 21; 23; 27].

In this continuing quest for a richer quantitative model, we also introduce a tool for exploring interactions of the simple and complex features LKT provides for the learner. The idea of explicitly including statistical interactions in logistic regression learner appears to be new and is related to the issue of multiple knowledge component labeling (where performance for an item has 2 or more KCs involved), which it can be used to model. When an item or step has multiple knowledge components [15] it implies that the prediction for that item or step depends on both in some way but the function for combination is never clear without expert prediction or empirical evidence. Interactions allow the modeling of conjunctive situations. If an interaction coefficient between the features for 2 KCs has a positive sign, that implies that the influence of the combination of skills is super-additive, which models a conjunctive relationship (where you need both KCs to do very well). In contrast, if the interaction coefficient is negative that implies a compensatory situation, where the combination is less than additive, implying the skills can in some way substitute for each other. Finally, the lack of an interaction (with interaction coefficient 0) implies that there are 2 independent routes to success, and being good at one route does not compromise or enhance your ability to perform given the other route. While these coefficients do not perfectly explain why the KCs combine, they are flexible in capturing many possible ways the skills may combine.

2. LOGISTIC KNOWLEDGE TRACING

2.1 Overview

Logistic knowledge Tracing (LKT) is an R package for creating learner models with logistic regression [25]. LKT is a formalized approach to creating logistic regression models that subsumes many other models and provides flexibility that allows better determination of an accurate model than off-the-shelf approaches like the Additive Factors Model (AFM), Instructional Factors Analysis (IFA), Performance Factors Analysis (PFA), PFA-Decay or Recent-PFA (R-PFA) [6; 8; 13; 14; 16; 22; 26], though it can replicate these models.

2.2 AutoKC

The new "autoKC" facility of LKT uses covariance clustering, which is a method to find item relationships using data on how starting KC performances covary. The first step of this method is to convert the performance for each participant into the raw probabilities for each of the KCs for each participant and sum these results. Missing values are replaced by the mean for the KC. This large matrix is normalized to logits and screened for outliers and then multiplied by its transpose to produce a square covariance matrix that provides a row/column for each starting KC that characterizes how its performance covaries across participants relative to all the other items. The final step is to cluster these covariance vectors (we use partial medoids here, using the pam R package, with default Euclidean distances), and then label the data with a new column that represents the cluster membership.

Developed by Pavlik, Cen, Wu, and Koedinger [23], covariance clustering is a method to describe how each item or existing KC in a domain model is related to all other items or KCs (using a measure of conditional log odds to represent covariance). This method computes a vector for each item representing the conditional probability table for success and failure for the items/KCs relative to all other items/KCs. The pairwise relationships between each vector are similar to the relationships inferred in POKS (Partial Order Knowledge Spaces, [9; 10]), a method related to Falmagne's work [11; 12]. An advantage of covariance clustering is that it characterizes each pairwise relationship between items/KCs in terms of the relationship with all other items/KCs. Pavlik et al. [23] used clustering to establish how to group items by using this KC/item relational vector as a set of features. We updated this method in a recent publication [20]. AutoKC has the advantage of sharing the speed of clustering and matrix multiplication methods, so it provides very few disadvantages beyond the additional complexity of implementation.

2.3 Applying autoKC in the LKT

Application of AutoKC in LKT requires the specification of how the autoKC column is used as a predictor. Here we apply autoKC in the context of simple (1 coefficient for slopes of successes for each KC and 1 coefficient for failures) and full (coefficients for all KC for successes and failures) versions of Performance Factors Analysis (PFA). We add versions of autoKC to these models and also apply autoKC by itself with only student and KC intercept values to model basic student knowledge and KC difficulty.

First, we investigate autoKC by itself. In this case, autoKC can be used to specify any number of clusters between 2 and the total number of KCs minus 1. A value of 2 would mean that autoKC was used to relabel all of the KCs as belonging to cluster 1 or cluster 2. Using this new "autoKC" with an LKT feature (such as the PFA success or failure count features), means that the effect of successes will now apply within cluster, so a success for cluster 1 will affect all items/KCs in cluster 1. In contrast, a value of N-1 (1 less than the actual count of KCs) means that the most similar two KCs will be clustered, and all the other KCs that remain will be in cluster with only 1 of the previous KCs (i.e., they are not clustered).

Then we investigate autoKC added as an addition to the PFA model for the original KCs. In this case, we have the PFA model, but we also have additive cluster-level effects. So, if k (number of clusters) is set to 2, like in the prior example, it basically says that while there are KCs there are also relationships between the KCs (the autoKC). A value of 2 indicates there are two broad groups of KCs (perhaps they are word problems and numeric problems) that share some base aptitude (for example, one group may depend on a reading comprehension skill and the other math skill).

2.4 Applying interactions in LKT

We also apply interactions among the KCs. In this introduction to interactions in logistic knowledge tracing, we test a basic variant of PFA which, instead of only counting the success and failures for each KC, also counts the successes and failures for students (faculty KC) and for the autoKC. We look at these features additively, but we see the most advantage in various interaction situations. To simplify this analysis, we always look at the interactions of success (or failure) for the KC with autoKC or with the student (faculty KC). We only look at these 2 interactions:(successes for KC X successes for autoKC or faculty and failures for KC X failures for autoKC or faculty.

We apply these interactions in the context of autoKC, but the interactions tool is independent of the autoKC tool. With an interaction it is possible to explore a new space of logistic regression models that may capture student data better than the simple addition of terms. In particular, interactions may also be excellent to capture the differential predictive accuracy of features in different practice contexts or to combine features to look at how levels of one feature (e.g., retention) are moderated by another feature (e.g. practice quantity).

2.5 Model contrasts

To understand the function, effect, and/or lack of effect of autoKC and component interactions, we have selected a number of possible models of practice and compared them. This helps us see the relative effect of each of the model configurations so that we can understand the autoKC and interaction functions in LKT. The nomenclature for these models is as follows. All of the models are **log PFA** corresponding to the hypothesis that there are decreasing marginal returns to 1+successes or 1+failures according to a natural log function [7; 18]. **IRT** corresponds to a baseline without PFA terms (only containing KC difficulty and student ability). **Faculty** corresponds to a very simplistic theory of knowledge where each student has 1 KC for all items, this serves as a baseline as it was used in prior research [16]. **autoKC** refers to a case where a KC is composed by the autoKC algorithm. **Simple** and **full** are described in Section 2.3. When a model combines KCs with autoKCs (by having 2 success predictors and 2 failure predictors (the final 6 models) these combinations are listed as **additive** or **interactive**

3. DATASETS

3.1 MATHia

The Carnegie Learning MATHia Course 2 dataset was CMU Datashop dataset #4845, which was a sample of 500 students from the larger MATHia 2019-2020 dataset at the same location. MATHia Course 2 is part of a middle school math curriculum. This data was from the Modeling Two-Step Expressions workspace and included 119,379 observations. The 9 knowledge components in this workspace are shown in Tables 1 and 2.

3.2 Cloze

The cloze statistics cloze dataset was Memphis Datashop #1465 and included 58,316 observations from 478 participants who learned statistical concepts by reading sentences and filling in missing words. Participants were adults recruited from Amazon Mechanical Turk. There were 72 KCs in the dataset (some KCs with different feedback conditions were combined since the content was identical), derived from 18 sentences, each with 1 of 4 different possible words missing (cloze items). The number of times specific cloze items were presented was manipulated, as well as the temporal spacing between presentations (narrow, medium, or wide). The post-practice test (filling in missing words) could be after 2 minutes, 1 day, or 3 days (manipulated between students). The stimuli type, manipulation of spacing, repetition of KCs and items, and multiple-day delays made this dataset appropriate for evaluating model fit to well-known patterns in human learning data (e.g., substantial forgetting across delays, benefits of spacing).

4. MODEL COMPARISONS

The reader should keep in mind that the number of possible models in LKT is practically limitless, so selecting comparisons was done with a focus on improving understanding of the possibilities rather than looking for the best fit model. We do learn about the knowledge models for both datasets, in addition to the properties of autoKCs and interactions, but much of this knowledge is qualitative in this first analysis of these features. While we do cross-validate our comparisons results, it seems true that the methods here are very highly dependent on the domain analyzed (as the results show). Nevertheless, we do see consistent improvements in both model complexity and fit that will make these methods important to investigate further.

Table 1. autoKC assignments for k=2 and 3.

Model	Equation
IRT	$\text{intercept}_i + \text{intercept}_k$
Faculty	$\text{intercept}_i + \text{intercept}_k + \text{successes}_i + \text{failures}_i$
Log Full autoKC	$\text{intercept}_i + \text{intercept}_k + \$\text{successes}_{aKC} + \$\text{failures}_{aKC}$
Log Simple PFA	$\$\text{intercept}_i + \$\text{intercept}_k + \text{successes}_k + \text{failures}_k$
Log Full PFA	$\text{intercept}_i + \text{intercept}_k + \$\text{successes}_k + \$\text{failures}_k$
Log Full PFA full autoKC additive	$\text{intercept}_i + \text{intercept}_k + \$\text{successes}_k + \$\text{failures}_k + \$\text{successes}_{aKC} + \$\text{failures}_{aKC}$
Log Full PFA Faculty additive	$\text{intercept}_i + \text{intercept}_k + \$\text{successes}_k + \$\text{failures}_k + \$\text{successes}_i + \$\text{failures}_i$
Log Simple PFA Faculty interactive	$\text{intercept}_i + \text{intercept}_k + \text{successes}_k + \text{failures}_k + \text{successes}_i + \text{failures}_i + \text{successes}_k\text{successes}_i + \text{failures}_k\text{failures}_i$
Log Simple PFA full autoKC interactive	$\text{intercept}_i + \text{intercept}_k + \text{successes}_k + \text{failures}_k + \$\text{successes}_{aKC} + \$\text{failures}_{aKC} + \text{successes}_k\$\text{successes}_{aKC} + \text{failures}_k\$\text{failures}_{aKC}$
Log Full PFA simple autoKC interactive	$\text{intercept}_i + \text{intercept}_k + \$\text{successes}_k + \$\text{failures}_k + \text{successes}_{aKC} + \text{failures}_{aKC} + \$\text{successes}_k\text{successes}_{aKC} + \$\text{failures}_k\text{failures}_{aKC}$
Log Simple PFA simple autoKC interactive	$\text{intercept}_i + \text{intercept}_k + \text{successes}_k + \text{failures}_k + \text{successes}_{aKC} + \text{failures}_{aKC} + \text{successes}_k\text{successes}_{aKC} + \text{failures}_k\text{failures}_{aKC}$

Table 1 shows the model names and the equation notation. Intercepts refer to fixed coefficients for each level of the factor subscripted. i, k and aKC represent the indices for student, KC and autoKC. The $ is used to represent non-intercept additive factors which are fit with 1 coefficient for each level of the factor (i.e. 1 coefficient per KC) instead of a single coefficient for all levels.

4.1 Fit for different applications of autoKC with and without interacting KCs

4.1.1 MATHia results

Figure 1 shows the surprising conclusion first of all that the 9 KC domain model is not better than a simple faculty model. This doesn't imply that the KC model is inherently incorrect, because we can also see that the Full KC model with the faculty model added actually does substantially better. So, the main implication here is that there is some additional student domain learning beyond just the independent KCs. Figure 1 shows no particular advantage for autoKC, but the simple PFA simple autoKC interaction model does do particularly well given its small number of parameters (just 6 added to the IRT model).

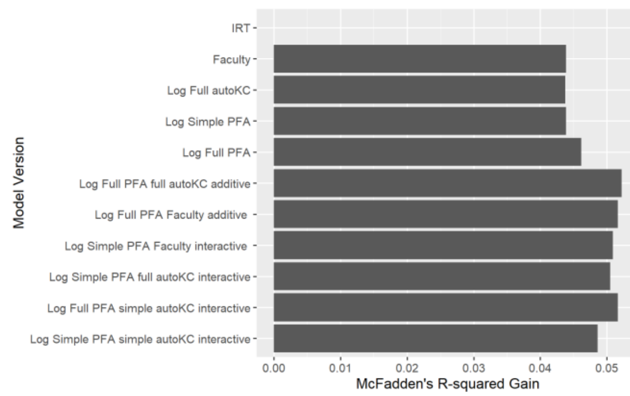

Figure 1. IRT is a baseline of .1726 R^2 using 5-fold unstratified crossvalidation. Graph shows improvement in crossvalidated R^2 values. autoKC set k=2.

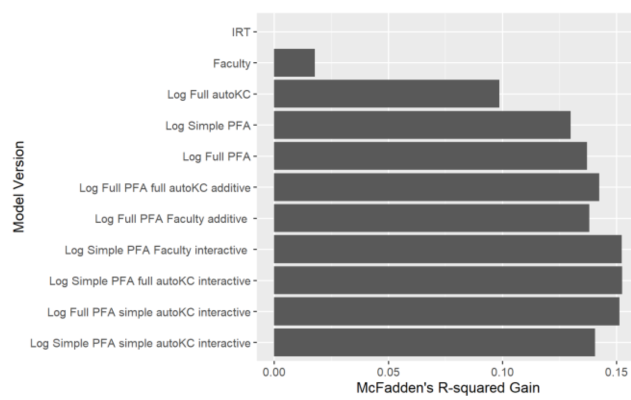

Figure 2. IRT is a baseline of .1581 R^2 using 5-fold unstratified crossvalidation. Graph shows improvement in crossvalidated R^2 values, autoKC set to 40.

4.1.2 Cloze results

For the cloze data in Figure 2 we see a somewhat different pattern. While it is clear the faculty (1 KC) model is better than IRT, it isn't very good even compared to the equally parameterized (1 parameter for success and 1 for failures in both case) simple PFA model, which uses the 72 KCs. This implies strong independence of the items which we also see confirmed in Figure 5. When autoKC was set to 40 KCS we see that the fit for 72 KCs is not approached, which shows the independence of the KCs again. Again, we see that the simple PFA simple autoKC interaction model does particularly well given its small number of parameters. We suspect this is due to the how the interaction terms provide flexibility in the shapes of the learning curves.

4.2 Fit for all levels of k for each dataset

4.2.1 MATHia results

Figures 3 and 4 show respectively the results for different K for the 3rd (Log Full autoKC) and 6th models in Figures 1. The Figure 3 result suggest no benefit for the fit for the Log Full autoKC model at any k, the number of clusters. Comparing Figure 1 Log Full KC shows an advantage for the original model with 9 KCs at R^2= .02191.)

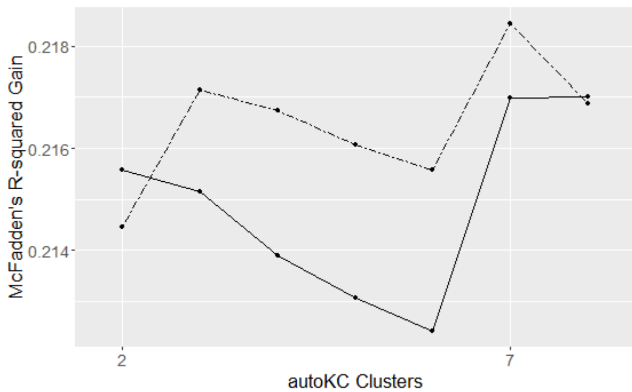

Figure 3. Crossvalidated fit for Log Full autoKC model as function of k. Dashed line is a randomly sampled autoKC comparison. Solid line is the 3rd model.

In contrast, when autoKC is added on top of the 9 KCs we do see some indication that 2 or 3 autoKC clusters provides an additive advantage on top of Log Full PFA.

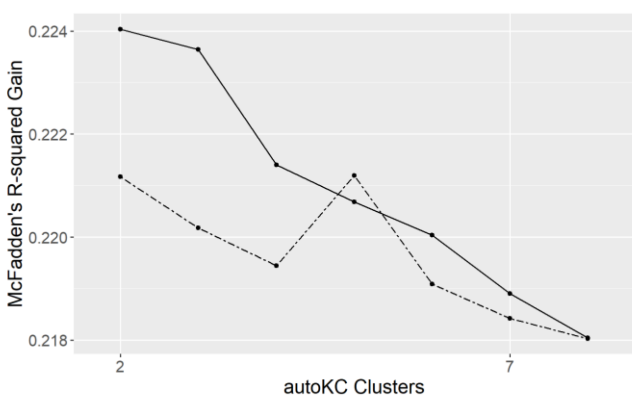

Figure 4. Crossvalidated fit for Full PFA with additive autoKC as a function of k. k=2 or k=3 provides an optimal advantage over random. Dashed line is a randomly sampled autoKC comparison. Solid line is the 6th model.

In a case like this is interesting and perhaps useful for understanding the model to see how autoKC is grouping the skills by inspecting the autoKC assignment matrix (which is available from the LKT R function). Table 2 presents the assignment matrices found for 2 and 3 clusters in Figure 4. On the surface it seems the groupings are meaningful, but it would take further analysis of the MATHia system to draw strong conclusions, furthermore these results are likely to be noisy unless very large datasets are used as input.

Table 2. autoKC assignments for k=2 and 3.

k=2	k=3	original KC
1	1	define variable-1
1	1	enter given, reading numerals-1
1	1	enter given, reading words-1
2	2	find y, any form-1
2	2	identifying units-1
2	3	write expression, negative intercept-1
2	3	write expression, negative slope-1
2	3	write expression, positive intercept-1
2	3	write expression, positive slope-1

4.2.2 Cloze results

Figure 5 shows the third model Log Full autoKC across the 70 settings of k. We see that there is always loss of fit compared to the original KCs, but by comparing the random and model KC assignments lines in the figure we can see that starting around 7 KCs until about 27 KCs there is some indication that the clustered groupings are better than random. This figure suggests the items are highly independent since more clusters always helps.

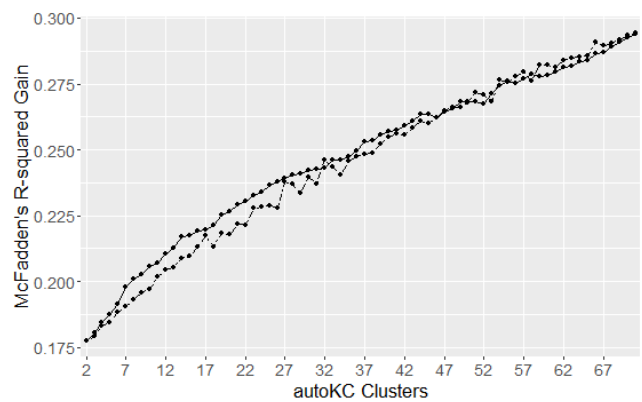

Figure 5. Crossvalidated fit for Log Full autoKC model as function of k. Dashed line is a randomly sampled autoKC comparison. Solid line is the 3rd model.

The pattern in Figure 6 shows more potential for the 6th model than Figure 5 does for the 3rd model, since we can see some advantages for the autoKC model terms when added to the PFA model. Lower values of k result in larger clusters with weaker associations between the items in the clusters, and thus less gain for each cluster. Middle values of k result in smaller clusters of more similar items resulting in more benefits. However, as k becomes very high the autoKC produces many singleton clusters and singleton clusters do not improve fit, since the input data becomes equivalent to Full PFA, which was already included in the model.

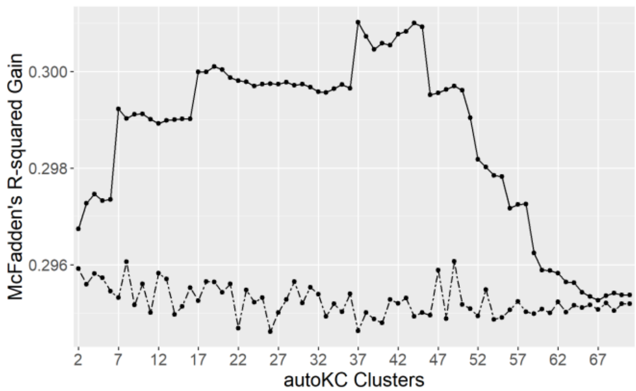

Figure 6. Crossvalidated fit for Full PFA with additive autoKC as a function of k. k between 37 and 45 provides an optimal advantage over random. Dashed line is a randomly sampled autoKC comparison. Solid line is the 6th model.

5. LIMITATIONS AND FUTURE WORK

The simplification of the models here was a limitation of the work, and also reveals future work to refine the usages of these new LKT methods. One simplification was that we used the PFA log of success and log of failures predictors for each KC and choose to use those as the basis to look at autoKC and interactions. On the other hand, consider that we could have computed autoKC for the logitdec feature (a recency weighted function of the logit of performance) and that would have been an elegant way to model each of the clusters as a knowledge domain and track aptitude in that domain. Rather the current model we presented with log PFA predictors more closely represents a model of transfer within the cluster. Of course, there is also the limitation of covariance clustering itself, which is nondirectional, and could cluster either due to transfer of learning or due to shared aptitude (a third variable like prior knowledge might drive correlation).

Another limitation of this paper was our focus on introducing the two mechanisms together. While this paper nicely illustrates interesting interactions with autoKC, many interactions are possible, for example, interactions between group factors (such as the condition of an intervention experiment), student factors (such as the ethnicity or prior knowledge), item factors (such as word frequency or amount of practice) and trial factors (such as multiple choice or fill-in-the-blank/cloze). This limitation of the paper hides the enormous flexibility that interactions allow the modeler in LKT.

The models we showed also only had 1 or 2 layers of component predictors on top of the IRT terms. For example, our 2-layer models used the original KC layer and the autoKC layer. However, it is certainly possible to consider 3 or more KC layers using autoKC. For example, based on Figures 5 and 6 we might hypothesize that in addition to an autoKC k=40, we might also add a layer of much larger clusters with k=7. While there would certainly be diminishing returns from multiple layers, still creating a multilevel representation of the skill hierarchy seems likely to result in better predictions.

A final limitation is the curse of dimensionality due to the large feature sets in situations with high dimensionality. It isn't entirely clear this is an important issue since the feature spaces in our case will not be independent but tend to have patterns of correlations. Research has shown the curse of dimensionality may depend on independent features [28]. Nevertheless, future work will need to better examine the effect of dimension reduction prior to using clustering methods.

6. DISCUSSION

In summary, we saw incremental small improvements in fit, sometimes with reductions in the complexity of the model. These results suggest that the models presented here are practical to use in knowledge tracing applications for adaptive learning software [24]. In particular, it was notable that the reduced parameter (simple PFA simple autoKC) has many fewer coefficients than the full PFA model despite showing a slight advantage in fit.

While the model fit benefit of autoKC is small, the implications for adaptive learning may be considerably greater due to the way the clustering implies relationships among items. One way to describe our result is to say that we have created a model where there will be transfer among KCs without any loss of model fit. This emphasizes the fact that the model of transfer is itself valuable to guide pedagogy. If we know that there are relationships between KCs, this can be used to guide the introduction of related KCs in a grouped fashion and also improve the adaptation of items within an autoKC cluster by using their correlation to guide inference about other items. If two KCs are in a cluster that means if performance is high on one, the other is also likely to be above average. Similarly, when KCs are clustered it may mean that if a student learns one of the KCs, they will also have improved their performance for the other KCs in that cluster. This sort of knowledge may be useful for adaptive learning systems by allowing the software to infer whether KCs are mastered or need practice even when the KC has not been practiced itself (only other items in its cluster having been practiced). Such information could be useful to guide the introduction of new content and the revisiting of old content in adaptive learning systems.

7. ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation Learner Data Institute (NSF #1934745) projects and a grant from the Institute of Education Sciences (ED #R305A190448).

8. REFERENCES

[1] Atkinson, R.C., 1972. Ingredients for a theory of instruction. *American Psychologist 27*, 10 (1972), 921-931. DOI= http://dx.doi.org/http://doi:10.1037/h0033572.

[2] Atkinson, R.C., 1972. Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology 96*, 1, 124-129.

[3] Barnes, T., 2005. The Q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, J. Beck Ed. AAAI Press, Pittsburgh, PA, USA, 39-46. DOI= http://dx.doi.org/http://doi:10.1.1.531.3631.

[4] Birenbaum, M., Kelly, Anthony E., and Tatsuoka, Kikumi K., 1992. *Diagnosing knowledge states in algebra using the rule space model.* Educational Testing Service.

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

[6] Cen, H., Koedinger, K.R., and Junker, B., 2007. Is over practice necessary? – Improving learning efficiency with the Cognitive Tutor through educational data mining. In *13th Int. Conf. Artif. Intell. Educ.*, Los Angeles, CA, USA, 511-518. DOI= http://dx.doi.org/http://doi:10.1.1.142.7340.

[7] Chi, M., Koedinger, K.R., Gordon, G., Jordan, P., and VanLehn, K., 2011. Instructional Factors Analysis: A cognitive model for multiple instructional interventions. In *4th International Conference on Educational Data Mining*, M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero and J. Stamper Eds., Eindhoven, The Netherlands, 61-70.

[8] De Boeck, P., Bakker, M., Zwitser, R., Nivard, M., Hofman, A., Tuerlinckx, F., and Partchev, I., 2011. The estimation of item response models with the lmer function from the lme4 package in R. *Journal of Statistical Software 39*, 12 (2011), 1-28. DOI= http://dx.doi.org/http://doi:10.18637/jss.v039.i12.

[9] Desmarais, M.C., Meshkinfam, P., and Gagnon, M., 2006. Learned student models with item to item knowledge structures. *User Modeling and User-Adapted Interaction 16*, 5, 403-434.

[10] Desmarais, M.C., Pu, X., and Blais, J.-G., 2007. Partial Order Knowledge Structures for CAT Applications. In *Proceedings of the 2007 GMAC Conference on Computerized Adaptive Testing* (2007).

[11] Falmagne, J.-C., Doignon, J.-P., Cosyn, E., and Thiery, N., 2003. The assessment of knowledge in theory and in practice. *Institute for Mathematical Behavioral Sciences Paper 26*.

[12] Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., and Johannesen, L., 1990. Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review 97*, 2 (Apr), 201-224.

[13] Galyardt, A. and Goldin, I., 2015. Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining 7*, 2 (2015), 83-108. DOI= http://dx.doi.org/https://doi.org/10.5281/zenodo.3554671.

[14] Gong, Y., Beck, J.E., and Heffernan, N.T., 2011. How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Journal of Artificial Intelligence in Education 21*, 1 (2011), 27-46. DOI= http://dx.doi.org/http://doi:10.3233/JAI-2011-016.

[15] Koedinger, K.R., Pavlik Jr., P.I., Stamper, J., Nixon, T., and Ritter, S., 2011. Avoiding Problem Selection Thrashing with Conjunctive Knowledge Tracing. In *Proceedings of the 4th International Conference on Educational Data Mining*, M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero and J. Stamper Eds., Eindhoven, the Netherlands, 91–100.

[16] Koedinger, K.R., Yudelson, M.V., and Pavlik, P.I., 2016. Testing theories of transfer using error rate learning curves. *Topics in Cognitive Science*(2016), published online. DOI= http://dx.doi.org/http://doi:10.1111/tops.12208.

[17] Lan, A.S., Studer, C., and Baraniuk, R.G., 2014. Quantized Matrix Completion for Personalized Learning. In *Proceedings of the 6th International Conference of Educational Datamining*.

[18] Lindsey, R.V., Shroyer, J.D., Pashler, H., and Mozer, M.C., 2014. Improving students' long-term knowledge retention through personalized review. *Psychological Science*(January 20, 2014). DOI= http://dx.doi.org/10.1177/0956797613504302.

[19] Pardos, Z. and Dadu, A., 2018. dAFM: Fusing psychometric and connectionist modeling for Q-matrix refinement. *Journal of Educational Data Mining 10*, 2 (2018), 1-27. DOI= http://dx.doi.org/http://doi:10.5281/zenodo.3554689.

[20] Pavlik Jr, P.I., Eglington, L., and Zhang, L., 2021. Automatic Domain Model Creation and Improvement. In *Proceedings of The 14th International Conference on Educational Data Mining*, C. Lynch, A. Merceron, M. Desmarais and R. Nkambou Eds., 672-676.

[21] Pavlik Jr., P.I., Cen, H., and Koedinger, K.R., 2009. Learning factors transfer analysis: Using learning curve analysis to automatically generate domain models. In *Proceedings of the 2nd International Conference on Educational Data Mining*, T. Barnes, M.C. Desmarais, C. Romero and S. Ventura Eds., Cordoba, Spain, 121–130.

[22] Pavlik Jr., P.I., Cen, H., and Koedinger, K.R., 2009. Performance factors analysis -- A new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, V. Dimitrova, R. Mizoguchi, B.D. Boulay and A. Graesser Eds., Brighton, England, 531–538. DOI= http://dx.doi.org/http://doi:10.3233/978-1-60750-028-5-531.

[23] Pavlik Jr., P.I., Cen, H., Wu, L., and Koedinger, K.R., 2008. Using item-type performance covariance to improve the skill model of an existing tutor. In *Proceedings of the 1st International Conference on Educational Data Mining*, R.S. Baker and J.E. Beck Eds., Montreal, Canada, 77–86.

[24] Pavlik Jr., P.I. and Eglington, L., 2021. The Mobile Fact and Concept Textbook System (MoFaCTS) Computational Model and Scheduling System. In *22nd International Conference on Artificial Intelligence in Education (AIED 2021) Third Workshop on Intelligent Textbooks* In CEUR workshop proceedings (Vol. 2895). 1-15.

[25] Pavlik, P.I., Eglington, L.G., and Harrell-Williams, L.M., 2021. Logistic Knowledge Tracing: A Constrained Framework for Learner Modeling. *IEEE Transactions on Learning Technologies 14*, 5, 624-639. DOI= http://dx.doi.org/10.1109/TLT.2021.3128569.

[26] Pelánek, R., 2017. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction 27*, 3 (2017), 313-350. DOI= http://dx.doi.org/http://doi:10.1007/s11257-017-9193-2.

[27] Sahebi, S., Lin, Y.-R., and Brusilovsky, P., 2016. Tensor Factorization for Student Modeling and Performance Prediction in Unstructured Domain. *International Educational Data Mining Society*.

[28] Zimek, A., Schubert, E., and Kriegel, H.-P., 2012. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal 5*, 5 (2012/10/01), 363-387. DOI= http://dx.doi.org/https://doi.org/10.1002/sam.11161.