Automatic Domain Model Creation and Improvement

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

ABSTRACT

We describe a data mining pipeline to convert data from educational systems into knowledge component (KC) models. In contrast to other approaches, our approach employs and compares multiple model search methodologies (e.g., sparse factor analysis, covariance clustering) within a single pipeline. In this preliminary work, we describe our approach's results on two datasets when using 2 model search methodologies for inferring item or KCs relations (i.e., implied transfer). The first method uses item covariances which are clustered to determine related KCs, and the second method uses sparse factor analysis to derive the relationship matrix for clustering. We evaluate these methods on data from experimentally controlled practice of statistics items as well as data from the Andes physics system. We explain our plans to upgrade our pipeline to include additional methods of finding item relationships and creating domain models. We discuss advantages of improving the domain model that go beyond model fit, including the fact that models with clustered item KCs result in performance predictions transferring between KCs, enabling the learning system to be more adaptive and better able to track student knowledge.

Keywords

Knowledge component model, domain model, learning transfer

1. INTRODUCTION

This paper describes preliminary progress to create a multimethod pipeline to determine the knowledge model (or domain model) that allows the most accurate prediction of performance in an adaptive learning system using a quantitative model of practice. A broad use of quantitative models of practice is to predict performance and make pedagogical decisions [1; 2]. To do this effectively, models typically assign sets of problems or items specific skill tags (often called knowledge components, or KCs). Having such an identification allows a system to monitor which skills have been learned and which need more practice. 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 for systems to make pedagogical decisions. Without such an assignment, a system would conceivably need to schedule all items for practice to ensure mastery, so the assignment or "domain model" must be accurate for a system to perform well. Improvements in the domain model may result in better pedagogical decisions in a system. This paper describes a more general approach to improve these critical domain models, a tradition that has included much prior work [3; 5; 14; 15; 19; 20; 22].

In addition to improving domain models, we highlight how these methods may alter how many quantitative models work by enabling models where multiple knowledge components can influence a single practice trial. While such models are not new [13], specifying them with experts is time consuming and error prone. Despite this difficulty, domain models that include the potential for multiple KCs affecting a single performance also typically capture transfer when a shared KC is used in multiple items. In addition to making models more accurate, this transfer has large potential impacts on pedagogy in a complex adaptive instructional system since transfer in an adaptive system means that a KC's performance may bias the selection of other items that share KCs. This transfer will occur because the shared KC will affect the item predictions, making items sharing a KC more or less likely to be practiced.

2. ANALYSIS METHOD

We have developed an automatic domain model improvement algorithm with a highly configurable analysis pipeline.

2.1 Step 1

First is the preprocessing stage. In this stage, some matrix-based method will produce some featural vector of information representing each item. There are two ways this method might process the data from an educational system, either all at once or sequentially in the order the student saw the items. In the first case, this would include methods such as SPARFA-Lite, which assumes one observation for each skill for each student [14]. Our example in this paper uses the SPARFA-Lite model and a simpler model based on covariance clustering [20]. For our examples, Step 1 meant averaging KCs performances for each subject to get a student performance by KC. More advanced methods such as tensor analysis can proceed with sequential data for each student. However, this is future work not presented here.

2.2 Step 2 Infer Feature Matrix

In this step, the method is applied to the data to get some matrix. Currently, the pipeline has two possibilities at this stage, but we plan to include multiple methods in future work as we look to our long-term goal of building a shareable tool for the EDM community.

2.2.1 Covariance Clustering

Developed by Pavlik, Cen, Wu, and Koedinger [20], 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

Philip I. Pavlik Jr., Luke Eglington and Liang Zhang "Automatic Domain Model Creation and Improvement". 2021. In: Proceedings of The 14th International Conference on Educational Data Mining (EDM21). International Educational Data Mining Society, 672-676. https://educationaldatamining.org/edm2021/
EDM '21 June 29 - July 02 2021, Paris, Franc

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, [7; 8]), a method related to Falmagne's work [10; 11]. 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. [20] used clustering to establish how to group items by using this KC/item relational vector as the set of features.

2.2.2 SPARFA-Lite

Developed at Rice University by Lan, Studer, and Baranuik [14], SPARFA is a factor analysis method to extract factors from binary-valued data. It provides an association matrix similar to a dAFM Q-matrix with graded associations of concepts with items. The "Lite" version simplifies the method by reducing the parameters and allowing automatic inference of the optimal number of concepts. This method works differently than dAFM, but it provides similar results, allowing for direct comparisons. Also, the ability to infer the optimal number of concepts may be a useful constraint when applying other algorithms.

2.3 Step 3 Cluster Principal Components of Features

In this step, the information matrix is clustered using some method to group items into clusters. Our current implementation first uses RSVD (Randomized Singular Value Decomposition) to simplify the information matrix. We see from the pattern in the results section how the quantity of RSVD components influences the clustering result. We are currently using K-means clustering for clustering, so our search is across both RSVD number of components (N) and K for the number of K means clusters.

For this step, we have also done considerable experimentation with the cmeans fuzzy clustering method, which provides a 0 to 1 index of how strongly each KC is associated with each cluster. Typically, we have used this by specifying a threshold (which can be optimized with search) over which an item belongs to each cluster or not. This assignment allows for membership in multiple clusters, which means that unlike the method in Step 2.4, the item is assigned to potentially many clusters. Typically, when we use this method, we have weighted the effect of prior practice for the KC clusters according to the number of KC clusters involved in a performance. This weighting is not necessary for the simpler K-means implementation since the added KC column only assigns each KC to 1 KC cluster.

2.4 Step 4 Fit with New Model

We used the new model as an overlay such that we created a column with the cluster id for each KC for each trial. This overlay procedure means that while the KC and clusters are independent, practicing an item may affect other items if they share a cluster. To do this, we first describe our starting model, which was simply PFA (Performance Factors Analysis, [17]) using the logarithms of practice counts for successes and failure (adding 1 to each to permit the logarithms). Where θ values are Student ability and KC difficulty respectively, and S and F represent the count of prior success and failures for the KC j for the student i.

$$logit(p_{ijt}) = \beta_1 \log_e S_{ij} + \beta_2 \log_e F_{ij} + \\ \theta_i + \theta_j$$

The new model was defined using cluster-id (c) as a KC in an additive compensatory model. Prior research suggests such compensation among KCs works well for prediction [6; 16].

$$logit(p_{ijt}) = \beta_{1\mathrm{j}} \log_e S_{ij} + \beta_{2\mathrm{j}} \log_e F_{ij} + \\ \beta_{3\mathrm{c}} \log_e S_{ic} + \beta_{4\mathrm{c}} \log_e F_{ic} + \\ \theta_i + \theta_j$$

Two versions of this model with clusters were compared; the first version was as described, and the second version was a control condition where the cluster column was sampled at random from the Q-matrix. This control condition should exhibit the same amount of overfitting due to adding parameters but none of the benefit of a coherent clustering solution. These models are compared using 2 to N components and 2 to K clusters by iterating to Step 3 to search a space of models.

In the context of our future work, we plan to allow users of our tool to specify candidate models with different configurations and terms using a logistic knowledge tracing R package freely available [18]. It is possible that different learner models may be implemented at this step since the Q-matrices we are creating may be used in many types of learner models.

2.5 Step 5 Splitting and Merging

Just as steps 3 and 4 may iterate to find optimal K and N, steps 4 and 5 may iterate to refine the model in Step 4. This step describes our future planning for a tool to optimize Q-matrix type models of knowledge domains.

Splitting takes the original KC model and uses the KC model from the clustered features to determine hypotheses for how KCs might be split. So if a KC in the original model is in 2 clusters, the model would test whether that that was best represented by the default model (include the effect of the cluster and the KC for each KC) or whether the cluster was unnecessary and the fit was just as good by splitting the KC into two different KCs and dropping the effect of the cluster KC. Further, we could also test whether the specific clusters proposed for each KCs even improves fit by removing them entirely as a third hypothesis. Two of these three possibilities correspond to Learning Factors Analysis (LFA) [5], and the third (including the cluster instead of using a split) advances the approach.

Merging uses the cluster model like LFA, but instead of splitting KCs, the clusters are used to evaluate three hypotheses about whether existing KCs can be merged into a single KC. One hypothesis is that the KCs are best represented as separate but influence each other through the shared cluster membership. The second hypothesis is that the specific cluster was unnecessary, and the two KCs should be merged into 1 KC. Finally, the third hypothesis is that the 2 KCs are separate and that the cluster predictor should be omitted.

Step 5 is similar to backward and forward stepwise regression methods, and so it is clear this method would be very likely to cause overfitting due to the way it will tailor the model term to capture the data iteratively. To prevent this problem, the solutions produced are robustly cross-validated. By tuning the model to maximize cross-validation accuracy, we aim to find quantitative thresholds for when to add or subtract terms from the model with a result that is efficient and parsimonious.

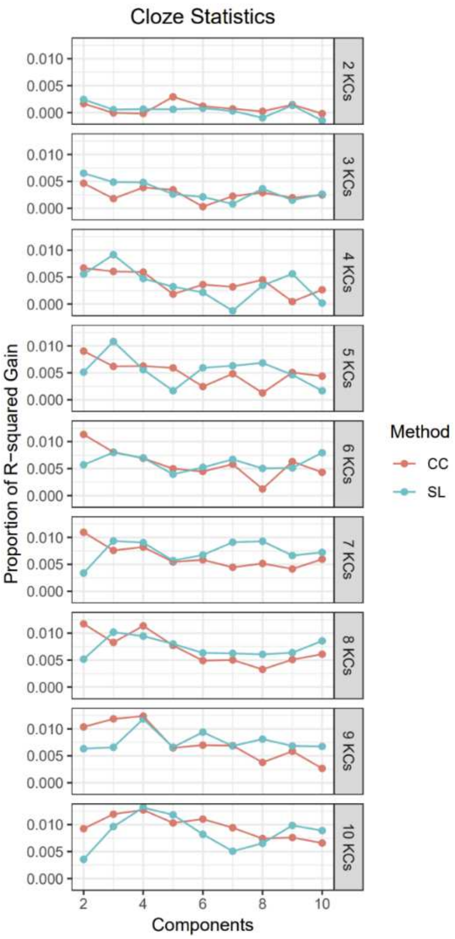

Figure 1. Changes in fit including differing numbers of additional clusters for Cloze dataset using covariance clustering (CC) or SPARFA-lite (SL).

3. DATASETS

The statistics cloze dataset 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 144 KCs in the dataset, derived from 36 sentences, each with 1 of 4 different possible words missing (cloze items). The number of times specific cloze items were presented was manipulated, and 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). The dataset was downloaded from the Memphis Datashop repository.

In the Andes dataset, 66 students learned physics using the Andes tutoring system, generating 345,536 observations. Participants were given feedback on their responses as well as solution hints. Additionally, participants were asked qualitative “reflective” questions after feedback (for more details, see [12]).

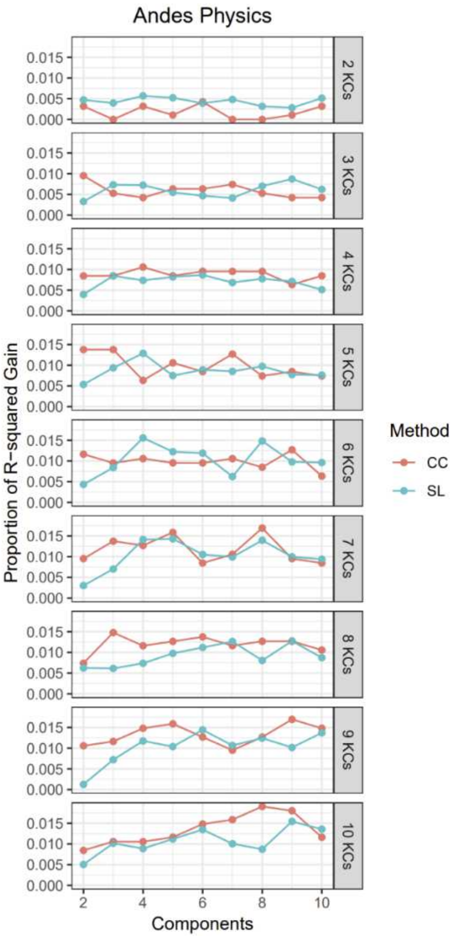

Figure 2. Changes in fit including differing numbers of additional clusters for Andes Physics data using covariance clustering (CC) or SPARFA-lite (SL).

4. RESULTS

Figures 1 and 2 show the result for the two datasets. The proportion of R-squared gain indicates the improvement in R-squared for the true clustered model compared to the random comparison R-squared model as a proportion of the random comparison R-squared model. Because of the result's preliminary nature, we have not been able to produce smoothed figures through cross-validation. However, the results consistently show beneficial effects. In general, both methods have similar accuracy.

Both methods can achieve similar improvements via different parameters. However, it does appear that the efficacy of the methods differs somewhat across datasets. Covariance clustering found the best solution in the Andes dataset, with SPARFA-Lite having the best solution in the Cloze dataset. This preliminary result suggests that applying multiple approaches to the same dataset may be advisable, especially when the underlying domain structure is unknown. Different domain modeling algorithms may differ in their ability to detect this underlying domain structure.

To understand better the results shown in Figures 1 and 2 we can query the model for the parameters for the cluster KCs to confirm that they are meaningful due to the structure of PFA. Normally we would expect the cluster KC coefficients for success and failure to be more different if the model was labeling real KCs since it is typically the case that successes predict future success more than failures. Indeed, test comparison shows exactly this pattern; for example, considering the model with 10 components and 10 KCs in for the Physics data with covariance clustering applied, we see the success is .56 higher than failure. In contrast, for the randomized model, the value was .12 higher for success than failure (best explained as the overfitting we might expect for such a mechanism).

5. DISCUSSION

In the present paper, we described our ongoing work to automate both the process of searching for domain models and the search method (e.g., covariance clustering vs. SPARFA-Lite). Many approaches have been proposed to infer domain [3; 5; 14; 15; 19; 20; 22], but there has been little comparison. However, comparison among approaches is important because their different underlying assumptions and limitations will interact with the learning domain's true underlying structure. For example, if the learning domain is calculus, various prerequisite skills from other branches of mathematics may be necessary (e.g., algebra, trigonometry). In other domains, learning one KC before another may enhance learning but not be required. Learning how to compute a sample's mean may facilitate learning to compute the *median* due to contrasting their different procedures. However, neither is a prerequisite to learn the other. Domains vary in the extent that learning one KC may transfer to another, and the researcher may not have strong theories a priori that could help constrain the KC model search. Thus, choosing one method with specific assumptions and limitations across different knowledge domains may be inadvisable and result in suboptimal KC model solutions.

5.1 Future Plans

5.1.1 Additional domain model methods

There are several methods we hope to include in the system to analyze student data to produce the inference matrix, for example:

dAFM - Developed at Berkeley by Pardos and Dadu [15] and shown to improve the Piech [21] deep knowledge tracing algorithm. This method is a deep learning model that uses backpropagation to infer a Q-matrix type representation with graded skill assignments instead of binary assignments. The authors show how the model is a continuous neural network generalization of the AFM model used in the LFA method [5].

Tensor factorization – We have also been working with implementations of tensor factorization. Tensors allow the solution to integrate multiple sources of data, including a representation of time in the sequence of practice.

These methods may be more accurate because they allow the representation of sequence to capture order effects in the model that may be due to learning. However, another view might be that it makes it more vulnerable to the selection effects that led to particular practice sequences. In other words, domain model search algorithms that are sensitive to effects over time may be more likely to incorporate artifacts due to pedagogical decision rules (e.g., “drop item from practice after N successes”). For instance, in systems in which items are dropped from practice after a few successes (e.g., Assistments), the sequential order and temporal spacing will be different than in practice schemes in which items are not dropped from practice (e.g., [9]). In short, domain model extraction from datasets that were generated by an adaptive learning system will be influenced by the decision rules inherent to that system.

5.1.2 An ensemble approach to address individual model search limitations

We also intend to allow multiple approaches to be allowed within a single KC model development pipeline. For instance, approaches like dAFM have shown promise to improve KC models but require an initial KC model. However, this apparent limitation is only a problem if the goal is to find a single approach that resolves the problem of KC modeling. Instead, the goal can be reoriented towards finding the best ensemble and ordering of approaches that can be used in order to develop an optimal KC model. As an example, an optimal KC model may be created by making an initial model with SPARFA-Lite, followed by a final model using dAFM. Requiring a starter model is only a limitation if complementary approaches cannot be combined.

5.1.3 Integrating with learner model development

Learner models and domain models are strongly interdependent but frequently developed and refined independently. This separation probably limits progress on both fronts. Using relatively simple learner models when searching for improved domain models may lead to misleading results if the chosen learner model does not accurately represent learning, forgetting, transfer, and other important learning factors. Similarly, developing learner models without considering the chosen KC model's plausibility may lead to spurious results. Recently, we developed a framework to facilitate learner model development named Logistic Knowledge Tracing [18]. We aim to integrate automated KC model search and refinement into the LKT framework.

5.1.4 Representing transfer does more than improve model fit

Representing transfer among KCs can have significant pedagogical consequences that will not be apparent from model fit metrics (e.g., reduced RMSE, increased AUC). For instance, imagine a student is learning three items (A, B, and C). If the domain model considers A and B to be related because they share KC, practicing item A will influence both *when* and *how much* B is practiced. Depending on the strength of the transfer, practicing A may result in B being practiced being, being practiced before C, being practiced after C, or not being practiced until much later when forgetting has occurred (if the learner model assumes forgetting). The entire order of practice may change.

Another issue is the efficiency effect of such transfer. Consider that if the three items are independent, students may practice all three as necessary for mastery. In contrast, if item A affects item B through a shared KC, it will increase or reduce the amount of practice needed for mastery of B, which can reduce costly overpractice. In short, accounting for transfer among KCs may greatly improve practice *efficiency*, which may not be apparent when comparing domain models in terms of model fit metrics (e.g., RMSE, AUC, AIC). Ultimately, comprehensive evaluation of new KC models requires simulations or experiments to determine their effects on how practice is scheduled within an adaptive learning system. This need comes from how a new KC model may interact with pedagogical decision rules (e.g., mastery learning) and learner models (e.g., BKT, PFA) within an adaptive learning system to change the *sequence* of practice (e.g., due to quantifying transfer among items differently). These changes to the sequence may have significant impacts on student learning.

6. 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).

7. REFERENCES

[1] Atkinson, R.C., 1972. Ingredients for a theory of instruction. *American Psychologist 27*, 10 (Oct), 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., 2008. Comparing two IRT models for conjunctive skills. In *Proceedings of the Proceedings of the 9th International Conference on Intelligent Tutoring Systems* (Montreal, Canada2008), 796-798.

[7] 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.

[8] 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).

[9] Eglington, L.G. and Pavlik Jr, P.I., 2020. Optimizing practice scheduling requires quantitative tracking of individual item performance. *npj Science of Learning 5*, 1 (Oct. 15,), 15. DOI= http://dx.doi.org/10.1038/s41539-020-00074-4.

[10] 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*.

[11] 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.

[12] Katz, S., Connelly, J., and Wilson, C., 2007. Out of the lab and into the classroom: An evaluation of reflective dialogue in Andes. *Frontiers in Artificial Intelligence and Applications 158*(Jun.), 425-432. DOI= http://dx.doi.org/http://doi:10.5555/1563601.1563669.

[13] 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.

[14] 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*.

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

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

[17] Pavlik Jr, P.I., Cen, H., and Koedinger, K.R., 2009. Performance factors analysis: A new alternative to knowledge tracing. In *14th International Conference on Artificial Intelligence in Education*, V. Dimitrova, R. Mizoguchi, B.D. Boulay and A. Graesser Eds., Brighton, England.

[18] Pavlik Jr, P.I., Eglington, L.G., and Harrell-Williams, L.M., 2021, preprint. Logistic Knowledge Tracing: A constrained framework for learner modeling. *arXiv.org*.

[19] 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.

[20] 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.

[21] Piech, C., Spencer, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L., and Sohl-Dickstein, J., 2015. Deep Knowledge Tracing. *arXiv preprint arXiv:1506.05908*.

[22] 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*.