Pavlik Jr., P. I., Cen, H., Wu, L., & Koedinger, K. R. (2008). Using Item-type Performance Covariance to Improve the Skill Model of an Existing Tutor. In R. S. Baker & J. E. Beck (Eds.), Proceedings of the 1st International Conference on Educational Data Mining (pp. 77-86). Montreal, Canada.

Using Item-type Performance Covariance to Improve the Skill Model of an Existing Tutor

Philip I. Pavlik Jr.[1], Hao Cen[2], Lili Wu[1], and Kenneth R. Koedinger[1]
{ppavlik, hcen}@andrew.cmu.edu, lili@cs.cmu.edu, and koediger@cmu.edu
[1]Human Computer Interaction Institute, Carnegie Mellon University
[2]Machine Learning Department, Carnegie Mellon University

Abstract. Using data from an existing pre-algebra computer-based tutor, we analyzed the covariance of item-types with the goal of describing a more effective way to assign skill labels to item-types. Analyzing covariance is important because it allows us to place the skills in a related network in which we can identify the role each skill plays in learning the overall domain. This placement allows more effective and automatic assignment of skills to item-types. To analyze covariance we used POKS (partial order knowledge structures) to analyze item-type outcome relationships and Pearson correlation to capture item-type duration relationships. Hierarchical agglomerative clustering of these item-types was also performed using both outcome and duration covariance patterns. These analyses allowed us to propose improved skill labeling that removes irrelevant item-types, clusters related types, and clarifies the optimal temporal ordering of these clusters during practice.

1 Carnegie Learning's Bridge to Algebra Cognitive Tutor

Our goal was to examine a large dataset (>9 million problems step performances) to determine a skill model that we could subsequently use to produce improvements to a Cognitive Tutor [1]. While the tutor had a skill model coded by human domain experts, we felt that it would be useful to develop alternative methods of coding skills that might be less vulnerable to the possibility of human error and bias. The dataset was provided by Carnegie Learning Inc. from the Bridge to Algebra Cognitive Tutor for the 2006-2007 school year [2]. This tutor works by providing a systematic coverage with 44 units of pre-algebra content each of which has several sections. These sections consist of a problem type, which is composed of several steps or "item-types" which repeat over a sequence of similar problems.

Data from this tutor included times of problem step actions and outcomes of step actions. Because, the human skill model was coded at this step level, we also choose to examine student performances at the step level (henceforth these individual steps in problems in sections in units will be called ***item-types***). By using this level of analysis we will be able to qualitatively compare the skill model that the tutor uses with the skill model suggested in our analyses.

2 Areas of improvement

Human coding and selection of skills in a tutor may introduce the following three possible problems which our datamining algorithm addresses. *As we discuss skills, it should be assumed that we are considering skills more generally as latent variables that* *may represent knowledge components important to a concept, rule application, or other learnable proficiency necessary for responding to an item-type.*

2.1 Problem of irrelevant skills

This issue refers to labeling an item-type as needing a skill despite it having a weak relation with other proficiencies that the tutor is focused upon. These weak relationships may be due to the fact that the action is either too simple (probability of success near to 1), too hard (probability of success near 0) or that the task is irrelevant to (independent of) the more complex related target proficiencies. In the case of simple tasks, the skill should not be labeled because the item-type should not be included in the tutor since performance is already at/near the desired level and time is wasted in further repetition. In the case of the difficult task we can suppose that some actions should not be included in a tutor because they result in poor learning and frustrate students. In both of these cases, our method will ignore these skills since a skill lacking variance cannot covary. In the case of an independent task it clearly makes little sense to include the item-type in the tutor unless the data suggests it is associated with a target skill of the domain.

2.2 Problem of skill redundancy

This is the issue of labeling two different item-types with two independent skill labels despite strong similarities in the skill involved. When a human decides to label a skill there is always the question whether the action is statistically equivalent with another action in the tutoring system being analyzed. Of course, this is a very difficult problem to judge since it depends on whether variation in the actions is different enough to produce difference in performances that are of practical significance. Because if this human experts may tend to label skills as different despite strong similarities. This leads to a model that tends to have poor intuitions about the how much practice to give a particular skill because when similar item-types are treated as independent the model can make no conclusions about how much practice to give the second one depending on performance with the first one. Thus, a tutor with redundant skills will be forced to give too little or too much practice because it is ignorant of the underlying skill overlap.

2.3 Problem of skill ordering

It is generally thought that a curriculum has a fixed order in which earlier skills form part of later whole skills or target performances, but this order may be difficult to identify. The importance of curriculum order may be due to a general benefit for training part skills before introducing whole skills. Wightman and Lintern [3] have described two ways of splitting a whole task into components tasks: segmentation and fractionation. Segmentation splits the whole task into sequential steps, and fractionation splits the whole task into time-shared parts. Optimal curriculum ordering may require some part-training in either case. With segmented skills, part-training may allow more efficient targeted training that can avoid already learned parts of the whole task. In the case of fractionated skills, part-training may provide the initial skill that enables the learner to successfully practice the whole skill. Because our method can identify prerequisites, we can hope that it will help us answer questions about the ordering of skills in a principled way.

3 Analysis methods

The analysis began with the creation of response correctness contingency tables (see Section 3.1) and response duration correlations (see Section 3.2) for all pairwise item-type combinations within each subject's data. Following the creation of these relationship matrices, simple agglomerative clustering iterated until no item-type (clusters) had a relationship correlation product (see Sections 3.2 and 3.3) greater than the clustering coefficient. At this point, the POKS tests were computed on the item-type clusters and the prerequisite order graph was created (one link was also removed by the requirement for $r > 0$ duration correlation between POKS linked item-type clusters).

The parameters were as follows: a clustering coefficient of 0.55, a p_c value of 0.35 (this value needed to be so high because of the strict correction of degrees of freedom described in Section 3) and a_c values of 0.15. These parameters were selected to limit the number of links and restrict the clustering to obtain a graph that would provide us examples we could analyze as discussed by Desmarais [4]. Additionally, before analysis, we removed students with less than 250 transactions, and we removed item-types with less than 2000 transactions. These filters reduced our dataset to 1073 students with a total of 1,099,642 outcomes (the quantity of response durations was slightly lower since we only accepted durations for trials with a successful outcome and discarded durations greater than 60s). The analysis was restricted to this subset, which included all transactions from 3 contiguous units of the 44 in the tutor, to make it manageable to show a graphic of the results. Larger numbers of units/item-types could not be made to fit on a single page.

3.1 Partial order knowledge structures (POKS)

To construct the graph describing the prerequisite relationships in the data, we used the POKS method, which is one way to characterize the theory of knowledge spaces. Knowledge spaces describe how the learn units (skills or item-types in this paper) in a domain are learned in a constrained order [5]. Knowledge spaces have been investigated by many researchers using different methods [6, 7].

To explain the POKS method, first, we introduces the notations following [4]:

$A, B, \ldots$ the upper case Roman letters denote two item-types in test
$A \Rightarrow B$, knowing how to solve item-type A correctly leads to solving item-type B correctly (we have reversed this arrow notation in Figure 1.)
$P(B \mid A)$, the probability of getting item-type B right, given A was right
$P(\neg A \mid \neg B)$, the probability of getting item-type A wrong, given B was wrong
p_c, the minimum probability that $P(B \mid A)$ and $P(\neg A \mid \neg B)$ need to hold
α_c, the error of the POKS tests, which may be set differently for different tests
$N_{A \wedge B}$, the number of times that students get A right and B right*

$N_{A \wedge \neg B}$, the number of times that students get A right and B wrong*

$N_{\neg A \wedge B}$, the number of times that students get A wrong and B right*

$N_{\neg A \wedge \neg B}$, the number of times that students get A wrong and B wrong*

* Because the independence of observations assumption of the statistical tests was strained when considering repetitions of the same item-type for the same student, these values were normalized by dividing the each by the total so that they summed to 1. The statistical tests then assumed a number of degrees of freedom equal to the number of subjects in each pairwise comparison. This correction is overly conservative, but provides an unbiased correction for the sometimes great between-subjects variability in the *N* of repetitions.

$CDFBinomial(x, n, p)$, the cumulative density function of a binomial distribution of n trials and p success probability

The idea of POKS is that if $A \Rightarrow B$ perfectly, we would expect $P(B \mid A) = 1$, $P(\neg A \mid \neg B) = 1$ and that the contingency table shows a lack of independence . In reality, due to noise and imperfect $A \Rightarrow B$, we would expect the above two equalities not to hold exactly. Thus we can setup tests such that if $P(B \mid A)$ and $P(\neg A \mid \neg B)$ are above some threshold p_c, we can have some confidence of $A \Rightarrow B$. Therefore, for there to exist a relationship between A and B three tests must succeed. The first two tests check that $P(B \mid A)$ and $P(\neg A \mid \neg B)$ are above some threshold p_c, given the allowed test error α_c. The third test verifies whether the conditional probability $P(B \mid A)$ and $P(\neg A \mid \neg B)$ are different from $P(B)$ and $P(\neg A)$.

Test 1 returns true if $CDFBinomial(N_{A \wedge \neg B}, N_{A \wedge B} + N_{A \wedge \neg B}, 1 - p_c) < \alpha_c$.
Test 2 returns true if $CDFBinomial(N_{A \wedge \neg B}, N_{\neg A \wedge \neg B} + N_{A \wedge \neg B}, 1 - p_c) < \alpha_c$.
Test 3 returns true if the 2*2 contingency table of $N_{A \wedge B}$, $N_{A \wedge \neg B}$ $N_{\neg A \wedge B}$ and $N_{\neg A \wedge \neg B}$ passes a χ^2 test with error rate α_c.

3.2 Clustering based on conditional log odds

As we can see by examining the tests, they rely on the contingency table that is tabulated for each pair-wise item-type comparison. These contingency tables create a covariance structure that "places" each item-type in the POKS graph relative to the other item-types. By reflecting on this we can see that if we want to cluster the items based on the similarity of the required proficiencies, which would imply they require the same skills, we need a distance metric for item-types that a) captures that two items co-vary and b) can cope with the fact that two items may not be equally difficult despite having the very similar covariance structures. Requirement *a* means we need a distance metric that captures the structure of the contingency tables for item-type X_1 as compared to the contingency tables for item-type X_2. Requirement *b* means that this metric probably should not capture the structure of the tables relative to the outcome of performance X_1 or X_2. Rather, we should describe a distance metric that is computed conditionally for those cases where X_1 or X_2 is a success or failure. Requirement b is important for the purpose here because the tutor introduces item-types in a fixed order. This fixed order means differences in average performance between item-types may be caused by learning. However, this difference in performance between item-types that represent the same skill should not greatly alter the contingencies given the response is a success or failure.

To do this comparison of the covariance structure it helps to consider the data for two item-types (X_1 and X_2) as being organized into two vectors of contingency tables describing these item-types relationship with all other possible item-types (Y_n). If we consider that each contingency table is organized with X frequency results for A item-type and Y frequency results for B item-type, then for each X_n by Y_n contingency table we individually computed 2 values: one for when X_n is a success and one for when X_n is a failure. In each of these 2 cases, the log odds of B vs. ~B frequencies is used to capture the strength of the odds B:~B on a continuous scale. Because these log odds do not capture the effect of frequency of A or ~A and only capture the relative frequency of B vs. ~B they are not reactive to learning of A that does not affect the patterns of B vs. ~B, nor are they reactive to difference in the *n* of observations of the B:~B results. Using this procedure we computed these 2 log odds (one for A and one for ~A) for each contingency table for each vector of contingency tables (X_1 or X_2).

At this point we can describe vectors of log odds values for each column item-type X_1 and X_2 (getting 2 values conditional on A and ~A for each item X_n by Y_n pair) and compute their Pearson correlation to determine the nearness of the two item-types in the knowledge space. To do this clustering we used a simple agglomerative hierarchical clustering to cluster item-types into a new grainsize which implies clustered items share the same performance requirements (skill). This new method shares similarities with correlation clustering methods that have proven useful for graph partitioning [7] and is described further in the next section.

3.3 Integrating duration covariance information

Previous work to understand the knowledge space has focused exclusively on how performance success or failure can be used to determine ordered structures. However, besides possessing success data, we also had data on the duration of each item-type performance. This data allowed us to compute pairwise duration correlations (*r* values) of the item-types that correspond with the POKS tests for each pairwise item-type relationship. While it was perhaps possible to use these correlations in some joint function with the strength of the result of the POKS tests for each pair, at this point we just used these values as an additional filter on which POKS implications we accepted as significant. For this paper we choose to exclude any $A \rightarrow B$ pairs where $r < 0$.

More importantly the duration correlation vectors created for each item-type were themselves correlated to produce values that represented the degree of similarity in the duration relationships between item-types. This statistic for each pair of item-types was multiplied by the correlation from the outcome based (log odds vectors) correlation above, and the item-type pair with the highest correlation product is clustered in each step of the simple agglomerative clustering. Clustering continues until the correlation of pairwise correlation vectors is above the clustering coefficient.

4 Results

Figure 1 shows the POKS graph obtained from this analysis and corresponds to the groupings in Table 1 which provides additional statistics to help interpret the results. The ovals in Figure 1 represent collections (or individual) item-types which were a function of the clustering procedure (also grouped in Table1). Item-type Label indicates the following information (probability correct_Unit name_section number_Skill ID number). The table also provides the average duration and total number of database observations (Rps—repetitions in 1000s) for each item-type. Colors indicate the majority unit membership for the grouped item-types, where LCM – least common multiple unit, GCF – Greatest common factor unit, and FracRep involves a visual and written fraction representation unit. Edge labels provide the average p_c value and the duration correlation r.

4.1 Irrelevant knowledge components

The analysis failed to find any covariance relationship for 17 of the item-types in the 3 units of the tutor. These knowledge components provide an example of how this method can be used to suggest proficiencies that do not covary with the other item-types in the tutor. These so-called irrelevant skills tend to be knowledge components with higher probability correct because in cases with higher probability correct there is less chance to get the examples of not A and not B that are needed to pass the 2nd binomial test. This means that these items are found to be irrelevant because they are so easy that it is less likely to detect how they influence other item-types even where such relationships might exist given a similar problem with higher difficulty.

While further analysis would be necessary to determine if these skills were truly irrelevant, the method has provided us with an initial hypothesis about which skills might be removed from the tutor so that time saved to spend on item-types with stronger relationships with the other tutor content.

4.2 Redundant knowledge components

The clustering of item-types indicates that the pattern of success contingency tables and the pattern of duration correlations were similar for these item-types such that if item-types X and Y are in a cluster it indicates they have similar relationships to the other item-types. By extension we can suppose that this similar place in the covariance structure suggests that performance for these item-types is constrained by the same skill. The fact that this clustering occurs suggests that the human coders used statistically irrelevant features to code the item-types. For example consider the green ovals in Figure 1. The right oval includes a variety of item-types that might be described as understanding the denominator, while the left oval includes item-types that deal with the numerator.

Table 1. Key for graph.

Item-type Label	Rps	Duration	Unit	Sect	ID	Skill description
0.97_FracRep_1_43,35	2k	4.8s	FracRep	1	43,35	Identify benchmark fraction, Identify Fraction using shape
0.91_FracRep_1_46	5k	7.3s	FracRep	1	46	Identify fraction associated with each piece of a horizontal bar
0.92_FracRep_1_35,34	22k	6.2s	FracRep	1	35,34	Identify Fraction using shape, Identify non-benchmark fraction
0.87_FracRep_1_40	5k	7.7s	FracRep	1	40	Identify fraction associated with each piece of a circle
0.85_FracRep_1_38	6k	8.1s	FracRep	1	38	Identify fraction associated with each piece of a square
0.91_FracRep_4_55	4k	5.4s	FracRep	4	55	Identify number of desired groups—construct
0.88_FracRep_5_56,34	4k	14.1s	FracRep	5	56,34	Identify fraction using number line, Identify non-benchmark
0.93_FracRep_3_53	7k	6.8s	FracRep	3	53	Identify fraction of desired items
0.96_FracRep_1_47	2k	5.6s	FracRep	1	47	Count number of shaded parts in square (discontiguous)
0.94_FracRep_1_42	2k	5.7s	FracRep	1	42	Count number of shaded parts in circle (contiguous)
0.95_FracRep_1_49	2k	7.0s	FracRep	1	49	Count number of shaded parts in circle (discontiguous)
0.95_FracRep_1_44	2k	5.6s	FracRep	1	44	Count number of shaded parts in horizontal bar (contiguous)
0.87_FracRep_1_37	3k	7.5s	FracRep	1	37	Count number of shaded parts in square (contiguous)
0.9_GCF_2_28	28k	9.5s	GCF	2	28	List factor of large number
0.94_FracRep_2_40	3k	6.7s	FracRep	2	40	Identify fraction associated with each piece of a circle
0.79_FracRep_3_52	4k	8.4s	FracRep	3	52	Identify number of total items
0.76_FracRep_6_61,59	6k	5.8s	FracRep	6	61,59	Represent non-benchmark, Represent fraction using num line
0.89_FracRep_6_58	5k	6.3s	FracRep	6	58	Identify fraction associated with each segment of a number line
0.7_FracRep_1_32	7k	8.8s	FracRep	1	32	Identify fraction associated with each piece of a vertical bar
0.96_FracRep_2_45	3k	24.0s	FracRep	2	45	Identify number of equal divisions (horizontal bar)
0.94_FracRep_2_33	3k	25.8s	FracRep	2	33	Identify number of equal divisions (vertical bar)
0.86_FracRep_2_41	3k	23.3s	FracRep	2	41	Identify number of equal divisions (circle)
0.93_FracRep_2_39	3k	23.6s	FracRep	2	39	Identify number of equal divisions (square)
0.88_FracRep_6_57	5k	18.3s	FracRep	6	57	Identify number of equal divisions (number line)
0.75_FracRep_5_57	7k	22.2s	FracRep	5	57	Identify number of equal divisions (number line)
0.83_FracRep_1_33	6k	23.2s	FracRep	1	33	Identify number of equal divisions (vertical bar)
0.91_FracRep_1_45	6k	23.6s	FracRep	1	45	Identify number of equal divisions (horizontal bar)
0.87_FracRep_1_41	6k	23.0s	FracRep	1	41	Identify number of equal divisions (circle)
0.85_FracRep_4_54	4k	21.0s	FracRep	4	54	Identify number of equal groups from fraction
0.93_FracRep_1_39	6k	22.6s	FracRep	1	39	Identify number of equal divisions (square)
0.93_LCM_1_21	3k	7.3s	LCM	1	21	Identify LCM - is product
0.85_LCM_1_25	5k	9.7s	LCM	1	25	Identify LCM
0.82_LCM_1_26	5k	8.5s	LCM	1	26	Identify LCM - one number multiple of other
0.91_GCF_1_30	8k	6.9s	GCF	1	30	Identify GCF
0.87_GCF_1_29	9k	7.0s	GCF	1	29	Identify GCF - one number multiple of other
0.91_LCM_1_23	145k	5.5s	LCM	1	23	List consecutive multiples of a number
0.77_LCM_2_21	5k	10.6s	LCM	2	21	Identify LCM - is product
0.7_LCM_2_26	7k	9.2s	LCM	2	26	Identify LCM - one number multiple of other
0.68_LCM_2_25	9k	11.4s	LCM	2	25	Identify LCM
0.69_GCF_2_29	10k	15.2s	GCF	2	29	Identify GCF - one number multiple of other
0.66_GCF_2_30	11k	15.2s	GCF	2	30	Identify GCF
0.56_GCF_2_31	48k	10.7s	GCF	2	31	Identify number of items in each group from GCF

Much of this clustering may be necessary because the human coders were instructed to code in as fine a grain as practical. This instruction led to different skills being coded depending on whether the stimulus was a vertical bar, horizontal bar, circle, square, or number line. In contrast, the clustering method lumped these skills together indicating they may be actually the same proficiency. By splitting these groups into separate skills the human coder delinked these proficiencies relative to the tutor's automatic scheduling mechanisms. So, for example, if a student does very well on these clustered item-types as they are introduced, it will not result in less practice for the other items that our analysis suggests are in the cluster. Therefore by proposing these clusters we can address learning of the concept more efficiently because we can model transfer between item-types that are controlled by the same underlying proficiencies. Modeling transfer between item-types allows us to know when a particular concept, skill or procedure has been mastered despite the fact that we may not have given a student examples of all the item-types in the cluster. (Also note that sometimes the human coders did repeat the same skill IDs for isomorphic item-types in different sections of the same unit. As we can see in Table 1, our clustering method tended to confirm these human skill labels by clustering these item-types. E.g. skill id 33 (and others) appear twice in the same cluster indicating that the model agrees with human coders decision to label these item-types with the same skill despite the fact that they are in different sections of the same unit.)

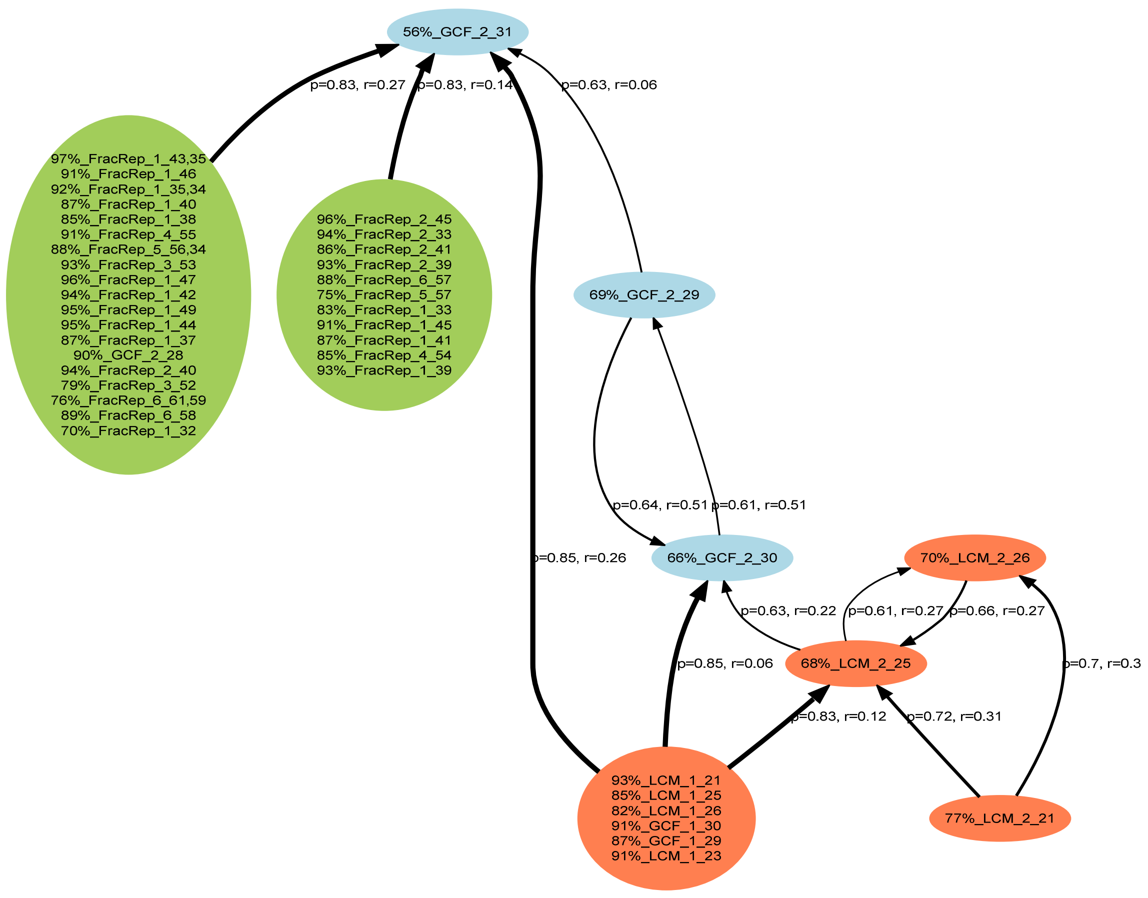

Figure 1. Graph structure described in the results section.

4.3 *Ordering of knowledge components*

The data comes from a tutor where the units follow a fixed order, and we can use our analysis to question the appropriateness of that order. As discussed in the introduction, we assume that introducing a prerequisite before its post-requisite will result in better learning, because each new idea will have been more adequately prepared by the scaffolding from prerequisite practice. This analysis of the optimal order is more difficult (than analysis of clustering or irrelevant skills) because the tutor repeats item-types, and learning caused by this repetition might explain why a downstream item-type performs better than an earlier item-type. However, while ideally we would include both orders of performance of any pair of item-types in our sample, it still seems safe to infer that very strong prerequisite relationships are not determined mostly by learning effects. Take for instance the position of the two green clusters (fraction concepts) relative to the GCF (greatest common factor unit)_2_31 item-type. Skill ID 31 involves a word problem in which students must produce the other factor for each of 2 products when the first factor has just been supplied by the student, e.g. "You have groups of 4 apples and 6 pears, what is the greatest number of equal sized groups of fruit you can make? (This is an ID 30 skill.) How many apples in each group? (ID 31) How many pears in each group? – (also ID 31)". This dependence of skill ID 31 in section 2 of the GCF unit on the fraction concept clusters seems plausible since this contextualized problem involves the denominator concept of understanding that wholes can be divided and also the numerator concept that these portions must be composed of a certain count of parts. While this reasoning might normally seemed strained, the support from the graph implies that the FracRep item-types should be practiced before this contextualized GCF section if we want to respect the recommendations of the theory of part-whole training to address the prerequisite skill first.

5 Conclusions

Future work will focus on integrating this knowledge space analysis with tracking of individual skills such as is currently used in the Bridge to Algebra Tutor. By integrating the knowledge space analysis it appears that we can get a rich perspective on what student actions might deserve to be coded as independent knowledge components. As we discussed in the results, this perspective should improve the performance of the model that tracks repetition of single skills in the tutor because that model can be modified to remove irrelevant skills, made less redundant by clustering skills, and made to better conform to the theory that prerequisites should be trained before later skills.

This integration may proceed as shown in work by Cen on the Learning Factors Analysis (LFA) method, which allows improvement in Cognitive Tutor models by searching a space of hypothetical skills for the combination that best fits previously collected data [8]. LFA starts with an initial cognitive model represented as a binary matrix that maps a collection of skills to each item-type (or item) and uses a set of customized item response models to evaluate the model fit produced by any given mapping of skills to item-types for a particular dataset. These binary matrices are based on the tentative judgments of human experts about the effect of the features of the item-types, and LFA can systematically incorporate those features into existing cognitive models by generating and searching for alternative skill labels as allowed for in the matrix. This method has been used by various researchers to evaluate cognitive models in geometry, physics and reading [9, 10] . However, the method still requires a domain expert to propose alternative labeling of skills along which the algorithm searches. The methods proposed in this paper show promising potential to combine the strengths of POKS, item-type clustering and LFA to answer various EDM research questions by allowing us to use POKS and item-type clustering to generate a starting or alternative binary skill matrix for LFA model search.

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] Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R. Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 1995, 4, p. 167-207.

[2] Carnegie Learning Inc. [Bridge to Algebra Cognitive Tutor data for the 2006-2007 school years].

[3] Wightman, D.C., Lintern, G. Part-task training for tracking and manual control. *Human Factors*, 1985, 27, p. 267-283.

[4] Desmarais, M.C., Maluf, A., Liu, J. User-expertise modeling with empirically derived probabilistic implication networks. *User Modeling and User-Adapted Interaction*, 1996, 5, p. 283-315.

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

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

[7] Desmarais, M.C., Gagnon, M.. Bayesian student models based on item to item knowledge structures. *First European Conference on Technology Enhanced Learning*, 2006, Crete, Greece.

[8] Cen, H., Koedinger, K., Junker, B. Learning Factors Analysis - A General Method for Cognitive Model Evaluation and Improvement. *8th International Conference on Intelligent Tutoring Systems*, 2006.

[9] Cen, H., Koedinger, K., Junker, B. Is Over Practice Necessary? – Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining. *13th International Conference on Artificial Intelligence in Education*, 2007, Los Angeles, CA.

[10] Rafferty, A., Yudelson, M.: Applying Learning Factors Analysis to Build Stereotypic Student Models. *13th International Conference on Artificial Intelligence in Education* (Best Student Paper), 2007, Los Angeles, CA.