Multiple Representations, Problem-Solving Behavior and Educational Outcomes

Ryan Carlson
Language Technologies Institute
Carnegie-Mellon University
Pittsburgh, PA 15213
rcarlson@cs.cmu.edu

Konstantin Genin
Department of Philosophy
Carnegie-Mellon University
Pittsburgh, PA 15213
kgenin@andrew.cmu.edu

Martina Rau
Human-Computer Interaction Institute
Carnegie-Mellon University
Pittsburgh, PA 15213
marau@cs.cmu.edu

Richard Scheines
Department of Philosophy
Carnegie-Mellon University
Pittsburgh, PA 15213
scheines@cmu.edu

Clark Glymour
Department of Philosophy
Carnegie-Mellon University
Pittsburgh, PA 15213
cg09@andrew.cmu.edu

ABSTRACT

We analyze log-data generated by an experiment with Mathtutor, an intelligent tutoring system for fractions. The experiment was designed to compare the educational effectiveness of instruction with single and multiple graphical representations. We extract features from the log-data characterizing the error-making and hint-seeking behaviors of each student. Using a latent class model, we cluster the students by their problem-solving profiles. We then explore how this profile interacts with mode of representation and post-test outcomes.

Four natural classes emerge from our analysis. We find that while graphical representation condition and post-test outcome are unconditionally dependent, problem-solving profile screens off representational condition from outcome for all but one class. This class of students is characterized by relatively high rates of error as well as a marked reluctance to seek help. It is in this sub-population that we see the greatest educational gains from multiple representations. The behaviors that characterize this group illuminate the mechanism underlying the effectiveness of multiple representations and suggest strategies for tailoring instruction to individual students. Our methodology can be implemented in an on-line tutoring system to dynamically tailor individualized instruction.

1. INTRODUCTION

Multiple Graphical Representations (MGRs) are used extensively in middle-school fraction instruction. Fractions can be alternately presented as pie and rectangle graphs, number lines, or discrete sets of objects. The educational psychology literature suggests that multiple-representations support learning in a variety of ways, though the experimental results are not univocal [1]. In particular, the mechanisms through which multiple representations influence student achievement are not well understood [2]. MGRs are generally implemented to achieve transfer because students who can translate between representations are creating deep knowledge structures that can impact learning outcomes.

Because user interaction with intelligent tutoring systems (ITSs) generates such a significant amount of data, these systems are well-suited for conducting experiments on the effect of multiple representations on learning outcomes. The proliferation of data also suggests that machine learning methods are relevant to the investigation of MGR effectiveness and the factors mediating their success. An active area of research within the ITS community revolves around supporting students within the environment. This involves automatically detecting a behavior and implementing an intervention on the fly. Work in this area ranges from detecting students who drill down through an ITS's hint sequence and find other ways to "game" the system [4] to providing hint support by identifying a lack of metacognitive hint-seeking behavior [3].

Prior work conducted on middle-school students working with ITSs found that multiple representations, in conjunction with self-explanation prompts, contribute to improved learning outcomes [7]. Subsequent studies examining error-rate, hint-use and time-spent in ITS logs failed to identify variables that mediate the effectiveness of multiple representations [6]. The mechanisms by which multiple representations improve learning outcomes remain poorly understood.

We conjecture that previous efforts to identify mediating factors were frustrated by heterogeneity in the problem-solving habits and behaviors of the student population under investigation. Using latent class analysis (LCA), a mixture modeling technique, we cluster students by their patterns of interaction with the tutor. We note that LCA and other latent modeling techniques are often used in the context of item response theory (IRT) [8], but we clarify that this is not the context under which we are employing our modeling technique.

Four natural classes emerge from our analysis. Two of the classes are characterized by a low propensity to seek help from the tutor. In one of these the students are simply confident: they make few errors, solicit little help and don't seem to need any. In the other they are stubborn: they make a relatively large number of mistakes but make little use of the support mechanisms the tutor provides. A third class is highly interactive: they make many mistakes and seek assistance readily. Students in the fourth class occupy a middle ground between the interactive and the stubborn: they make an average number of mistakes and will eventually seek help when they are having trouble.

We proceed to explore how multiple representations affect post-test outcomes. Confirming previous results, we find that experimental condition correlates positively with post-test outcome at the population level. We then explore the effect of multiple representations int the sub-populations. We first establish independence between cluster membership and experimental condition. This suggests that we are detecting pre-existing learning profiles, rather than artifacts of the experimental setup. Most interestingly, we discover that conditioning on cluster membership induces independence between experimental condition and outcome for all but the "stubborn" students. It is for these students that multiple representations remain strongly correlated with positive learning outcomes. We conjecture that these students lack the meta-cognitive skills to judge when their learning strategy is failing. These students are the most sensitive to pedagogical decisions since they are the least equipped to structure and manage their own learning.

Section 2 of what follows describes the initial experiment and elaborates on the differences between the representational conditions. We describe our feature extraction process and modeling decisions in Section 3. Section 4 summarizes the results of the model estimation and statistical analysis of the effects of multiple representations at the population and sub-population levels. We make concluding remarks and suggest profitable future directions in Section 5.

2. EXPERIMENT

In the Spring of 2010, Rau conducted an experiment wherein 290 4th and 5th grade students worked with an interactive fractions tutor for about 5 hours of their mathematics instruction. Students were randomly assigned to one of five experimental conditions. Conditions varied by the frequency with which they would be presented with a new fraction representation. Students in the Single representation condition worked exclusively with either a number line, a circle or a rectangle. Students in the Fully Interleaved condition would see a different representation with every new problem. Students in the intermediate conditions would go longer before they were presented with a different representation.

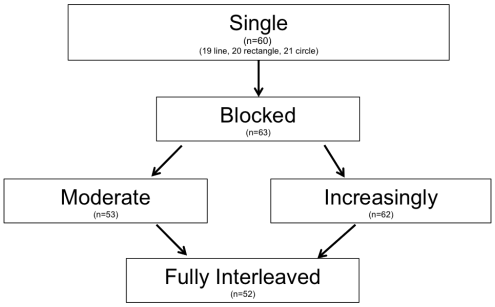

Figure 1: A partial ordering of experimental conditions by the frequency with which a new representation is presented.

When interacting with different graphical representations of fractions, students were able to drag-and-drop slices of a pie graph, for example, into separate areas. They were also able to experiment with changing the number of subdivisions in each graphical representation. Students received a pre-test on the day before they began working with the tutor and an immediate post-test on the day after they finished. Students also took a delayed post-test a week after the first. Previous investigation found that students in the multiple representation conditions significantly outperformed students in the single representation condition on the delayed post-test [6].

3. METHOD

Our analysis proceeds in three stages. Extracting features characterizing error and hint-seeking behavior, we transform the longitudinal log data into a cross-sectional form, with one observation per student. We then run Latent Class Analysis to identify sub-populations of students, using AIC and BIC to select the number of latent classes.

Once we have clustered our students, we investigate the interaction between the latent classes and their experimental conditions. We construct a contingency table binning the experimental conditions into the clusters estimated by the latent class model. We then run a Chi-squared test for independence between experimental condition and latent class. Chi-squared tests are also run to investigate dependence between latent class and post-test outcome and the conditional dependence of outcome and experimental condition, given latent class membership.

3.1 Extracting Features

The Cognitive Tutor captures a detailed log of each student's interactions with the tutor. It stores a time series of correct and incorrect answers, hint requests, interface selections and durations between interactions. Previous analysis (Scheines, Rau, 2012) extracted the average number of errors made per step, the average number of hints requested per step, and the average time spent per step from the log data. Similarly, we include the average number of hints requested (hints_req) and number of errors (num_errors) made per *problem* by each student. We also extract the average number of bottom-out hints (NumBOH) per student per problem – this is the average number of times a student exhausts the available hints in a given problem. We also note that it is not always the average of these features that best

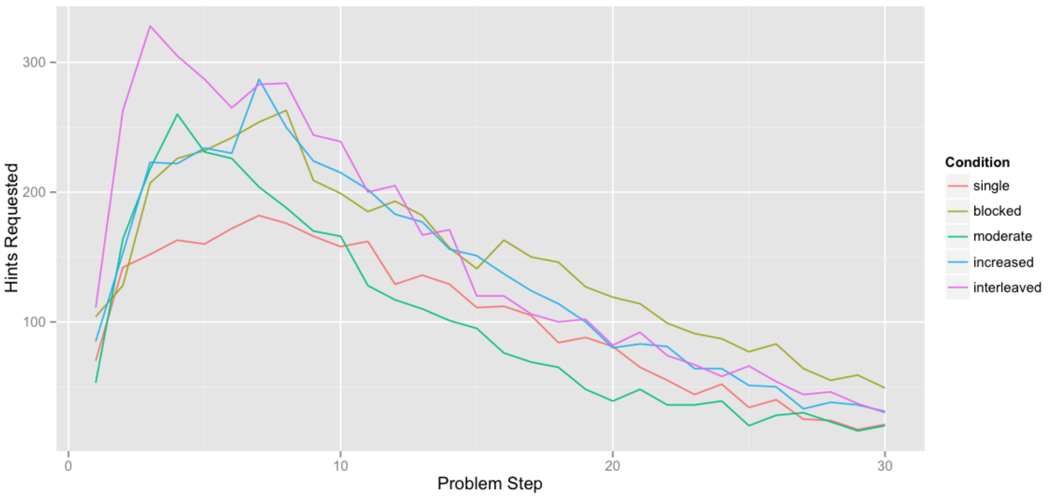

Figure 2: The x-axis represents the n_{th} interaction with the tutor across all problems. The y-axis is the total number of hints requested at the n_{th} step.

characterizes a student. For example, examination of the distribution of hints requested per step across experimental condition, shows a telling picture.

Note that students who received only one representation start out requesting the fewest hints, but students in the moderate condition eventually need fewer. Such considerations motivated our interest in the temporal distribution of hint behavior at the student level. We fit geometric distributions to the number of steps taken before the first hint request (FIRSTHINTGEOM) and to the number of errors before the first hint (STUBBORNGEOM). The estimated parameter is used to characterize the student's hint-seeking propensity in general and hint-seeking propensity when faced with adversity. For example, students in the first quintile of STUBBORNGEOM seek help soon after making a mistake, whereas students in the fifth quintile don't change their hint-seeking behavior even after making a large number of errors. Students in the first quintile of FIRSTHINTGEOM are likely to request hints early in a problem, whereas students in the fifth quintile are unlikely to request hints at any point.

3.2 Latent Class Analysis

Latent Class Analysis (LCA) is a modeling technique that determines subtypes based on multinomial distributions. We use LCA to categorize students into *latent classes* using discretized versions of the features described above. Table 1 shows summary statistics and cut-off points for the extracted features. The model maps a set of observed categorical variables onto a set of inferred latent variables.

We note that the categorical nature of the model has the potential to add some noise, since we must select numeric cutoffs to transform our variables into nominals. However, categorical models can offer greater interpretability by allowing us to organize our data into a small set of variables, which forms the basis for categorizing students into a small set of meaningful homogenous groups. Furthermore, it is not unreasonable to suspect that our variables are in some sense "truly" categorical.

The formal representation of LCA begins with $j = 1 \dots J$ observed variables, where each such variable j has a set of response variables $r_j = 1, \dots, R_j$. Each student has a distinct response pattern $y = (r_1, \dots, r_j)$.

Now we need to consider the latent classes. Let L be a latent variable with latent classes $c = 1, \dots, C$. Furthermore, let γ_c be the probability of membership in class c. Note that latent classes are exhaustive and mutually exclusive, so each student is a member of exactly one latent class. We also need to define the item-response probability $\rho_{j,r_j|c}$, which is the probability of response r_j to observed variable j, conditional on membership in latent class c. Each student provides exactly one response alternative to variable j. Given these constraints, note that

$$\sum_{c=1}^{C} \gamma_c = 1, \quad \sum_{r_j=1}^{R_j} \rho_{j,r_j|c} = 1.$$

Now that we have defined key variables, we can define the probability of observing a particular response vector based on the γ's and ρ's:

$$P(Y = y) = \sum_{c=1}^{C} \gamma_c \prod_{j=1}^{J} \prod_{r_j=1}^{R_j} \rho_{j,r_j|c}^{I(y_j=r_j)} \qquad (1)$$

where the indicator function $I(y_j = r_j)$ equals 1 when the response variable $j = r_j$. The parameters γ_c and $\rho_{j,r_j|c}$ are estimated by EM. Since EM is sensitive to starting probabilities, we pick the maximum likelihood over twenty-five runs. LCA is very similar to other EM-based algorithms. In fact, LCA is an application of multivariate mixture estimation using categorical variables with an additional local independence assumption, meaning that the observed variables are independent of each other conditional on the latent variable. This is a simplifying assumption similar to the one made in Naive Bayes; without it Equation 1 would have to be much more complicated. There is some work on relaxing this independence assumption [5]. To run latent class analysis, we

Table 1: Summary Statistics for Variables Used in Clustering

	mean	sd	median	min	max	20%	40%	60%	80%	100%
hints_req	0.78	1.27	0.34	0	11.22	0.06	0.19	0.5	1.31	11.22
num_errors	2.21	1.27	1.92	0.34	8.39	1.15	1.7	2.18	3.19	8.39
firstHintGeom	0.35	0.27	0.27	0.04	1	0.13	0.2	0.33	0.57	1
stubbornGeom	0.36	0.21	0.31	0.07	1	0.19	0.27	0.38	0.47	1
NumBOH	0.04	0.08	0	0	0.62	0	0	0.01	0.05	0.63

used `poLCA`, a freely available R package[1].

Note that unlike some common clustering algorithms (e.g., k-means), LCA produces "fuzzy" clusters–probability distributions over features for each class. To cluster students we identify their most likely class:

$$\arg\max_{c} P(Y = y \mid L = c) = \arg\max_{c} \gamma_c \prod_{j=1}^{J} \prod_{r_j=1}^{R_j} \rho_{j,r_j|c}^{I(y_j=r_j)} \tag{2}$$

We still need to fix C, the number of latent classes. We use two complexity-penalized log-likelihood scores to select an appropriate C: Akaike information criterion (AIC) and Bayesian information criterion (BIC). Plotting these statistics as we increment the number of latent classes, we look for a "knee" where both statistics either bottom-out or level off to identify the optimal value of C.

4. RESULTS

Figure 3 shows the parameter selection process described in section 4.2. Note that we chose to model four latent classes because BIC bottoms out and AIC levels off at that point.

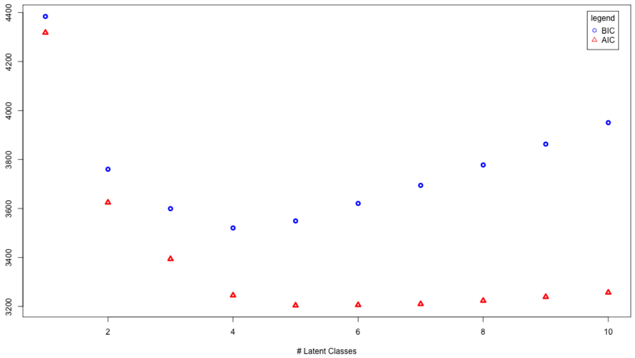

Figure 3: AIC and BIC over increasing number of latent classes. BIC bottoms out and AIC levels off at four classes, so we conclude that four latent classes best fits the data.

4.1 Exploring the Latent Classes

After selecting the appropriate C parameter, we extract membership probabilities for the individual students. Given a latent class, we know the probability distribution over each feature, and use Equation 2 to identify the most likely class for each student.

The feature distributions over each class are represented graphically in Figure 4. Note that each feature is listed along the horizontal x-axis, the value each variable takes takes is along the front-to-back y-axis, and the probability that the feature takes that value is given along the vertical z-axis. For example, consider the *hints_req* feature (average hints requested per problem) in Class 2. In that class, with high probability, students requested many hints (i.e., the highest categorical value for hints) per problem on average. As another example, students in Class 1 are more likely to make a moderate number of errors, though other error levels also occur with nontrivial probabilities. Note that lower values of *firstHintGeom* and *stubbornGeom* indicate a steep geometric slope, corresponding to a higher hint-seeking propensity.

How do we interpret latent class membership? Students in Class 1 are "Moderate", they ask for a moderate number of hints, make a moderate number of errors, and are moderately responsive to the interface. Students in Class 2 are "Interactive", they make a lot of errors, but respond by requesting many hints. These students are proactive in asking for help and are not shy about using the resources the cognitive tutor makes available. Students in Class 3 are "Confident", they don't ask for hints, but they don't seem to need them. Finally, the students in Class 4 are "Stubborn", they are fairly mixed in error-profile but they don't respond to mistakes with hint-requests. These students are not using all the resources that the cognitive tutor makes available.

4.2 Condition and Outcome

We construct a measure of student improvement and knowledge consolidation using the pre-test and the delayed post-test:

$$\text{Adjusted Delayed Post-Test Score} = \frac{\text{post\_test} - \text{pre\_test}}{1 - \text{pre\_test}}$$

We then construct terciles of the Adjusted Delayed Post-Test Score and run a Chi-squared test for independence of outcome from experimental condition. Confirming previous results, we reject independence at a p-value of .024 (See Table 3).

[1] `http://userwww.service.emory.edu/~dlinzer/poLCA/`

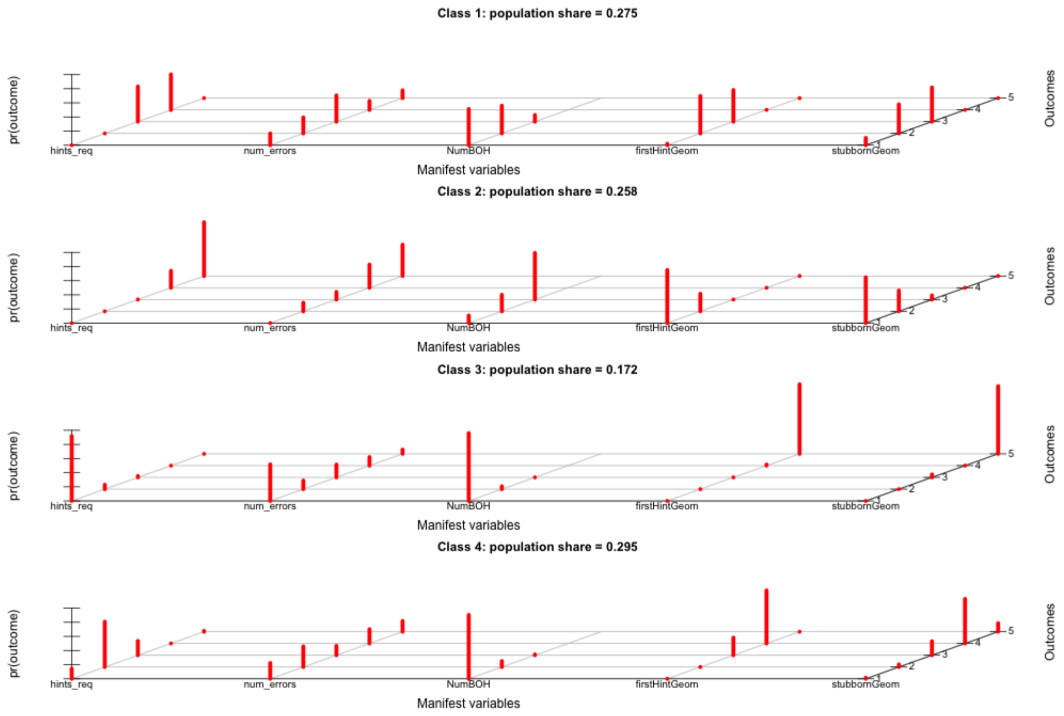

Figure 4: Visualization of feature distributions for each latent class. The left-to-right x-axis identifies each feature, the front-to-back y-axis identifies which value that feature takes, and the top-to-bottom z-axis describes the probability that the feature takes the value. Thus, given a feature and a class, the z-axis also describes the probability distribution over that feature in that class.

4.3 Cluster Membership and Outcome

We run a Chi-squared test for independence of latent class membership from outcome on the delayed post-test score. We reject independence at a p-value of .0075 (See Table 2). The problem-solving behaviors encoded by latent class membership are highly relevant to a student's learning outcome.

Table 2: Latent Class by Tercile of Adjusted Delayed Post-Test Score

	33%	66%	99%
LC 1	20	35	29
LC 2	33	26	14
LC 3	13	15	22
LC 4	31	20	32

$\mathcal{X}^2 = 17.52$, df = 6, p-value = **0.0075**

4.4 Condition and Cluster Membership

We may worry that experimental condition is inducing latent class membership. If this were the case, we would not be detecting pre-existing student profiles, but detecting an artifact of the experimental design. Using the Chi-squared test, we fail to reject independence at a p-value of .38 (See Table 3). It is therefore likely that our latent classes are detecting genuinely different student profiles, independent of experimental condition.

4.5 Condition, Outcome and Cluster Membership

Finally, we ask whether we still detect a dependence of outcome on condition within the latent classes we have defined. Interestingly, we find that latent class membership screens off condition from outcome for all classes but the fourth (See Table 4). Recall that these students rarely requested hints, even when they encountered difficulty. Our results suggest that since these students are not taking full advantage of the other resources the tutor makes available, it is the multiple representation condition that most strongly determines their learning outcome. The problem-solving behaviors of students in the other classes makes them insensitive to experimental condition.

4.6 Class Stability

See Figure 5

5. CONCLUSION & FUTURE WORK

We estimated a latent class model to classify students into four groups based on their error-rates and hint-seeking behaviors. We detected dependence of experimental condition and post-test outcome only in the class of students characterized by high-error rate and low hint-seeking propensity. That is, students who did not take full advantage of the resources that the Mathtutor offered were the ones most strongly affected by experimental condition. These students may not have the meta-cognitive skills required to know when to seek hints [3]. Our methods could be used by intelligent tutoring systems designers to detect students with this profile in real time. Tutoring systems could then intervene to target these students with multiple representations and to scaffold their hint-seeking behaviors. Future research into the mediating mechanisms of multiple representations could leverage our results to identify the relevant student sub-populations to investigate.

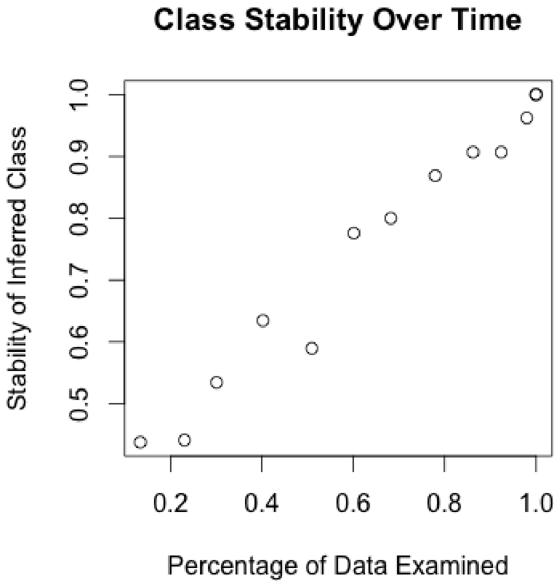

Figure 5: TODO

Table 3: Condition by Tercile of Adjusted Delayed Post-Test Score (left) and Latent Class (right).

	33%	66%	99%
blocked	14	29	20
increased	22	20	20
interleaved	13	21	18
moderate	18	13	22
single	30	13	17

$\mathcal{X}^2$ = 17.65, df = 8, p-value = **0.024**

	LC 1	*LC 2*	*LC 3*	*LC 4*
blocked	13	15	10	25
increased	21	16	10	15
interleaved	17	18	7	10
moderate	18	10	12	13
single	15	14	11	20

$\mathcal{X}^2$ = 12.85, df = 12, p-value = 0.38

Table 4: Condition and Tercile of Adjusted Delayed Post-Test Score, by Latent Class

LC 1	33%	66%	99%
blocked	2	8	3
increased	4	9	8
interleaved	4	9	4
moderate	4	5	9
single	6	4	5

$\mathcal{X}^2$ = 8.08, df = 8, p-value = 0.43

LC 2	33%	66%	99%
blocked	7	6	2
increased	9	5	2
interleaved	5	8	5
moderate	7	2	1
single	5	5	4

$\mathcal{X}^2$ = 6.95, df = 8, p-value = 0.54

LC 3	33%	66%	99%
blocked	0	5	5
increased	3	3	4
interleaved	2	2	3
moderate	3	4	5
single	5	1	5

$\mathcal{X}^2$ = 7.41, df = 8, p-value = 0.49

LC 4	33%	66%	99%
blocked	5	10	10
increased	6	3	6
interleaved	2	2	6
moderate	4	2	7
single	14	3	3

$\mathcal{X}^2$ = 17.4837, df = 8, p-value = **0.025**

6. REFERENCES

[1] S. E. Ainsworth. The functions of multiple representations. *Computers and Education*, 33(2-3):131–152, September 1999.

[2] S. E. Ainsworth. Deft: A conceptual framework for learning with multiple representations. *Learning and Instruction*, 16(3):183–198, 2006.

[3] V. Aleven, B. McLaren, I. Roll, and K. R. Koedinger. Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, 2006.

[4] R. Baker, A. Corbett, I. Roll, and K. R. Koedinger. Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction*, 2009.

[5] J. A. Hagenaars. Latent structure models with direct effects between indicators: Local dependence models. *Sociological Methods and Research*, 1988.

[6] M. Rau and R. Scheines. Searching for variables and models to investigate mediators of learning from multiple representations. *International Conference on Educational Data Mining*, 2012.

[7] M. A. Rau, V. Aleven, and N. Rummel. Intelligent tutoring systems with multiple representations and self-explanation prompts support learning of fractions. *International Conference of Artificial Intelligence in Education*, pages 441–448, 2009.

[8] W. J. van der Linden and R. K. Hambleton. *Handbook of Modern Item Response Theory*. Springer, 1997.