Improving Construct Validity Yields Better Models of Systematic Inquiry, Even with Less Information

Michael A. Sao Pedro, Ryan S.J.d. Baker, and Janice D. Gobert

Learning Sciences and Technologies Program, Worcester Polytechnic Institute
{mikesp, rsbaker, jgobert}@wpi.edu

Abstract. Data-mined models often achieve good predictive power, but sometimes at the cost of interpretability. We investigate here if selecting features to increase a model's construct validity and interpretability also can improve the model's ability to predict the desired constructs. We do this by taking existing models and reducing the feature set to increase construct validity. We then compare the existing and new models on their predictive capabilities within a held-out test set in two ways. First, we analyze the models' overall predictive performance. Second, we determine how much student interaction data is necessary to make accurate predictions. We find that these reduced models with higher construct validity not only achieve better agreement overall, but also achieve better prediction with less data. This work is conducted in the context of developing models to assess students' inquiry skill at designing controlled experiments and testing stated hypotheses within a science inquiry microworld.

Keywords: science microworlds, science inquiry, inquiry assessment, behavior detector, educational data mining, construct validity, feature selection, J48

1 Introduction

Feature selection, the process of pre-selecting features before running a data mining algorithm, can improve the performance of data mining algorithms (cf. [1]). Several automated approaches exist for finding optimal feature sets such as filtering redundant features [2], conducting heuristic searches (cf. [3]), using genetic algorithms [4], and clustering [5]. These procedures, though powerful, may yield sets that domain experts would not intuitively expect to align with the target class (construct). An alternative is to select features that specifically improve models' construct validity.

This alternative is motivated by our prior work in developing automated detectors of two scientific inquiry behaviors, designing controlled experiments and testing stated hypotheses, within a science microworld [6]. To build them, we first filtered features that correlated highly with each other, and then constructed J48 decision trees. The resulting detectors worked well under student-level cross-validation. However, upon inspecting them more closely, we noticed some features considered theoretically important to the constructs [7], [8], [9] were eliminated at the filtering step. Also, other features without theoretical justification remained. We believe this feature selection process may have yielded a feature set that did not represent all aspects of the behaviors, which in turn may have negatively impacted their predictive performance.

Thus, we explore in this paper whether selecting features with the goal of increasing a model's construct validity and interpretability can also improve a model's predictive ability. We do so by comparing two types of detectors for each behavior. One type is built with an automated feature selection strategy used in our original detectors [6]. The other type is built using a combination of manual selection and statistics to select successful features that theoretically align more closely with the behaviors.

We compare the predictive performance of the two types of detectors against a held-out test set in two ways. First, we compare the detectors' ability to predict behavior at the level of a full data collection cycle. This enables us to measure how well the detectors can be used for assessing performance, or for identifying which students need scaffolding when they claim to finish collecting data. In addition, it is useful to have detectors that can identify a student's lack of skill as quickly as possible so the software can "jump in" and support the student as soon as they need it to prevent frustration, floundering, or haphazard inquiry [10]. Thus, the second way we compare detectors is to determine how much student data is needed before inquiry behavior can be accurately predicted. The faster detectors can make valid inferences, the faster the system can help the students who need it.

2 Background and Datasets

2.1 Learning Environment and Behaviors of Interest

The Science Assistments Phase Change Microworld [6], [10], designed for use in middle school science classes, aims to foster understanding about melting and boiling processes of a substance via semi-structured scientific inquiry. A typical activity requires students to determine if one of four variables (like amount of substance) affects properties of a substance's phase change (like its melting point). Students address this goal by conducting inquiry in four phases: observe, hypothesize, experiment, and analyze data. Each one exercises different inquiry skills. The behaviors of interest for the analyses presented here, designing controlled experiments and testing stated hypotheses behaviors, occur in the experiment phase.

In the experiment phase, students collect data (trials) by designing and running experiments with a phase change simulation. Students can change the simulation's variable values, run, pause and reset the simulation, and view previously collected trials and stated hypotheses. Briefly, when collecting data, students design controlled experiments when they generate data that support determining the effects of independent variables on outcomes. They test stated hypotheses when they generate data with the intent to support or refute an explicitly stated hypothesis. More information about the microworld and constructs can be found in [6].

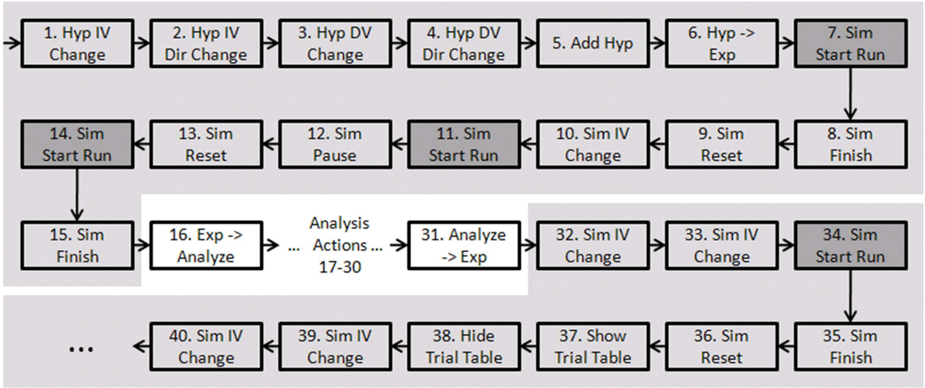


Fig. 1. Example sequence of student actions for a phase change activity. Two clips (shown in light grey) would be generated since the "Experiment" stage was entered twice.

2.2 Labeling Behaviors within the Learning Environment

The first step towards building detectors of these constructs, in both this paper and our previous work, was to employ “text replay tagging” of log files [6]. In this process, low-level student actions within microworld activities are extracted from the database. Next, contiguous sequences of these actions segmented into *clips* (see Figure 1). A clip contains all actions associated with formulating hypotheses (hypothesize phase actions) and designing and running experiments (experiment phase actions). We note that several clips could be generated for a single microworld activity since students could navigate through the inquiry phases many times, as shown in Figure 1. Clips are also the grain-size at which data collection behavior is labeled, and detectors are built.

Once clips are generated, a human coder applies one or more behavior tags to text replays, “pretty-prints” of clips. In this domain, a clip may be tagged as involving designing controlled experiments, testing stated hypotheses, both, or neither. Text replay tagging provides “ground truth labels” from which detectors of the two inquiry behaviors can be built. Next, we describe the datasets generated via text replay tagging student clips from which detectors will be constructed and tested.

2.3 Data Sets

Clips were generated from 148 suburban Central Massachusetts middle school students’ interactions within a sequence of four microworld activities. These clips were tagged to create the following training, validation and test data sets:

- *Training Set (601 clips).* Initially, two human coders tagged 571 clips for training and cross-validating the detectors in [6]. Since several clips could be generated per activity, a single, randomly chosen clip was tagged per student, per activity. This ensured all students and activities were equally represented in this data set. Interrater reliability for the tags was high overall (κ=.69 for designing controlled experiments, κ=1.0 for testing stated hypotheses). By chance, the stratification yielded few first clips, clips representing students' first data collection within an activity. To have a more representative training set, an additional 30 randomly selected first clips were tagged. In total, 31.4% of the clips were tagged as designing controlled experiments, and 35.6% as testing stated hypotheses.

- *Validation Set (100 clips).* A special set of clips was tagged by one human coder for engineering detectors with improved construct validity (described in more detail later). This set contained 20 randomly chosen first clips, 20 randomly chosen second clips, up through fifth clips. Clips were not stratified by student or activity. More stringent student or activity-level stratification was not used, because all students and activities were used to build the training set. Stratification would not remove biases already present in this data set. In total, 34.0% were tagged as designing controlled experiments, and 42.0% as testing stated hypotheses.
- *Held-out Test Set (439 clips).* A human coder tagged all remaining first through fourth clips in the data set for comparing detectors. This set did not contain fifth clips because only 2 remained in the tagged corpus. First clips in which one or no simulation runs occurred were also excluded, because demonstration of the inquiry behaviors requires that students run the simulation at least twice [10]. Such clips would trivially be identified as not demonstrating either behavior and could bias our comparisons. This set had 64.7% tagged as designing controlled experiments and 61.0% as testing stated hypotheses. Note that the data distribution of the behaviors was different in the held-out test set than the other data sets. This occurred due to random chance, but provides an opportunity to conduct stringent validation, since the base rates will be different in this data set than the other data sets.

Feature sets computed over clips, combined with text replay tags, form the basis for training and testing the detectors. Since the aim of this work is to compare models built from different feature sets, we discuss the feature generation and selection processes in more detail in the following section.

3 Feature Selection and Detector Construction

Our original designing controlled experiments and testing stated hypotheses behavior detectors considered 73 features associated with a clip [6]. Feature categories included: variables changed when making hypotheses, full hypotheses made, simulation pauses, total simulation runs, incomplete simulation runs (paused and reset before the simulation finished), complete simulation runs, data table displays, hypothesis list displays, variable changes made when designing experiments, and total actions (any action performed by a student). For each category, counts and timing values (min, max, standard deviation, mean and mode) were computed. In addition, the specific activity number associated with the clip was also included. A pairwise repeat trial count, the number of all pairs of trials with the same independent variable values [9], was also included, as was a unique pairwise controlled trial count, the number of non-repeated trials in which only one independent variable differed between them (cf. [7]).

All features were computed cumulatively, taking into account actions in predecessor clips, as in [6]. For example, given the actions shown in Figure 1, the total number of runs for clip 2 would be 5 (assuming no more runs had occurred after action 40).

We added five additional features to this set which seemed to have face validity as potential predictors of the two behaviors, giving a total of 78 features. In specific, we added *adjacent* counts for unique controlled trials and repeats. These are counts of successive trials (e.g. trial 2 vs. 3, 3 vs. 4) in which only one variable was changed (controlled) or all variables were the same (repeated). Since the controlled trials counts excluded repeat trials, we added two additional counts for controlled trials that did allow them, one pairwise and one adjacent. Finally, we added a feature to count when simulation variables explicitly stated in hypotheses were changed.

Two different approaches for feature selection over this set were employed to form behavior detectors. The first approach removed correlated features prior to building detectors (RCF detectors). The second approach involved selecting features geared at improving construct validity (ICV detectors). These procedures are discussed below.

3.1 Removed Correlated Features (RCF) Detector Construction

The original models in [6] were built in RapidMiner 4.6 as follows. First, redundant features correlated to other features at or above 0.6 were removed. Then, J48 decision trees, a Java-based implementation of C4.5 decision trees with automated pruning to control for over-fitting [11], were constructed. The RCF detectors of each behavior developed in this paper were built using this same process. However, they instead were built from the new feature set (78 features), and the enhanced training corpus.

The initial remove correlated features procedure eliminated 53 features. Of the 25 remaining features, 19 were timing values associated with the following feature classes: all actions, total simulation runs, incomplete simulation runs, simulation pauses, data table displays, hypothesis table displays, variables changed when making hypotheses, full hypotheses made, and simulation variable changes. The remaining 6 features were activity number and counts for the following feature classes: all actions, incomplete simulation runs, data table displays, hypothesis list displays, full hypotheses created, and adjacent repeat trials count (one of the new features added). RCF detectors for designing controlled experiments and testing stated hypotheses were then built based on this set of 25 features. Their performance will be discussed later in the Results section.

We note that this procedure eliminated some features which are considered theoretically important to both constructs. For example, counts for controlled trials, total simulation runs, and simulation variables stated in hypotheses changed were all filtered. These features are important, because they reflect theoretical prescriptive models of how data should be collected to support or refute hypotheses. Constructing controlled trials is seen as a key procedural component in theory on designing controlled experiments (cf. [7]). Similarly, running trials and changing values of the variables explicitly stated in the hypotheses both play roles in determining if hypotheses are supported. In addition, some features remaining did not immediately appear to map to theory on these constructs, such as the number of times that the student displayed the hypothesis viewer or data table. As discussed previously, we hypothesize these RCF detectors will not perform as well as detectors, because the remaining features do not theoretically align as well with the behaviors. Next, we describe how we selected features to yield detectors with improved construct validity (ICV detectors), which may in turn improve predictive performance.

3.2 Improved Construct Validity (ICV) Detector Construction

We selected features for the new detectors with increased construct validity (ICV) using a combination of theory and search. We first sought to understand how individual features related to the constructs. This was done by identifying which features had linear correlations to each behavior at or above 0.2. Several features did so with both behaviors: all actions count, total run count, complete run count, variable changes made when designing experiments, changes to variables associated with stated hypotheses when designing experiments, adjacent and pairwise controlled experiments counts (both with and without considering repeats), and pairwise and adjacent repeat trials counts. An additional feature correlated with designing controlled experiments, the number of simulation pauses. From this set of 11 features, the counts for controlled trials, repeat trials, and changing variables associated with stated hypotheses are all features used by others to directly measure procedural understanding associated with the behaviors [7], [8]. The other features, though not directly related, may also help distinguish procedural understanding. Thus, we kept all 11 features for the next round of feature selection.

From here, we reduced the feature set further by performing separate manual backwards elimination search (cf. [1]) for each construct as follows. Features were first ordered in terms of the theoretical support for them by a domain expert. Then, features were removed one at a time, starting with the one with the least theoretical support. From this candidate feature set, a decision tree was constructed using the training set. The resulting model's predictive performance was then tested on the *validation set* of 100 clips. If the candidate model yielded better performance than its predecessor, it was kept. If it did not, the candidate was rejected and another feature with low theoretical support was removed to form a new candidate set. This process was repeated, removing one feature at a time, until performance no longer improved.

Predictive performance was measured using A' [12] and Kappa (κ). Briefly, A' [12], the area under the ROC curve, is the probability that when given two clips, one labeled as demonstrating a behavior and one not, a detector will correctly identify which clip is which. An A' of 0.5 indicates chance-level performance, 1.0 indicates perfect performance. Cohen's Kappa (κ) assesses if the detector is better than chance at labeling behavior. κ of 0.0 indicates chance-level performance, 1.0 indicates perfect performance. When comparing two candidate models, the model with higher κ was preferred. However, if A' decreased greatly and κ increased slightly, the model with higher A' was chosen. If two models yielded the same values, the model with fewer features was chosen.

The best ICV detectors of each construct performed well over the validation set. The best designing controlled experiments ICV detector had 8 features (total run count and pause count were removed) and had A'=1.0 and κ=.84. The best testing stated hypotheses ICV detector had 5 features: variable changes made when designing experiments (both related and unrelated to stated hypotheses), unique pairwise controlled trials, adjacent controlled trials with repeats considered, and complete simulation runs. Its performance on the validation set was also strong (A'=.96, κ=.77).

4 Results: Comparing Predictive Capabilities of Detectors

Having created these two sets of detectors (RCF and ICV), we now can study whether selecting features more theoretically aligned with the two inquiry behaviors will yield better detectors than more traditional approaches. There are two key questions we address. First, which detectors predict best overall? Second, how quickly can detectors identify the two inquiry behaviors? Performance will be compared against the *held-out test set* only, rather than using cross-validation over all datasets. This was done for two reasons. First, the entire training set was used to select features for the ICV detectors. Using the full training set enabled us to understand the relationships between individual features and behaviors more thoroughly. Second, the search procedure for building ICV detectors likely overfit them to the validation set data.

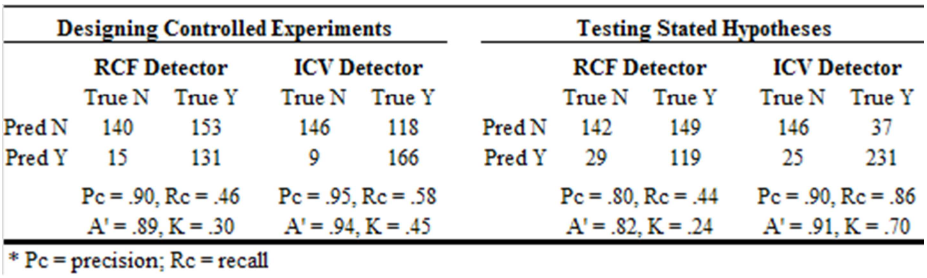
4.1 Comparing Detectors' Overall Performance

We compared detectors' performance at classifying behaviors in the held-out test set, labeled at the clip level. As a reminder, this comparison measures how well the detectors can be used for assessing performance, or identifying which students need scaffolding when they claim to be finished collecting data. Detectors are compared using A' and Kappa (κ). These were chosen because they both try to compensate for successful classification by chance [13], and have different tradeoffs. A' can be more sensitive to uncertainty, but looks at the classifier's degree of confidence; κ looks only at the final label, leading to more stringent evaluation. We note that statistical tests comparing models' A and κ are not performed. This is because students contribute multiple clips in the test set, and thus independence assumptions are violated. Meta-analytical techniques do exist to handle this (e.g. [14]), but our data did not have enough data points per student to employ them.

As shown in Table 1, the detectors with improved construct validity (ICV) detectors outperformed the removed correlated features (RCF) detectors within the held-out test set. For designing controlled experiments, both the RCF (A'=.89) and ICV (A'=.94) detectors were excellent at distinguishing this construct. However, the ICV detector was better at identifying the correct class (RCF κ=.30 vs. ICV κ=.45). Both detectors seem to bias towards labeling behavior as "not designing controlled experiments", as indicated by lower recall rates than precision rates (RCF recall=.46, precision=.90 vs. ICV recall= .58, precision=.95). This suggests that more students would receive scaffolding than necessary upon finishing data collection.

Upon inspecting the results for designing controlled experiments more closely, we noticed a large number of first clips with exactly two simulation runs had been mis-

Table 1. Confusion matrices and performance metrics for detectors' overall predictions.



Designing Controlled Experiments					Testing Stated Hypotheses				
	RCF Detector		ICV Detector			RCF Detector		ICV Detector	
	True N	True Y	True N	True Y		True N	True Y	True N	True Y
Pred N	140	153	146	118	Pred N	142	149	146	37
Pred Y	15	131	9	166	Pred Y	29	119	25	231
	Pc = .90, Rc = .46		Pc = .95, Rc = .58			Pc = .80, Rc = .44		Pc = .90, Rc = .86	
	A' = .89, K = .30		A' = .94, K = .45			A' = .82, K = .24		A' = .91, K = .70	

* Pc = precision; Rc = recall

classified. These kinds of clips comprised 26.7% of the held-out test corpus. When filtering these out (leaving 322 clips), the performance of the ICV detector was substantially higher (ICV A'=.94, κ =.75, recall=.83). The RCF detector's performance was also higher (RCF A'=.90, κ =.44, recall=.56), but did not reach the level of the ICV detector. The implications of this will be discussed later.

For the testing stated hypotheses behavior, the ICV detector again showed a substantial improvement over the RCF detector. The ICV detector was around ten percentage points better at distinguishing between the two classes (RCF A'=.82 vs. ICV A'=.91). Furthermore, κ and recall were much higher for the ICV detector than the RCF detector (RCF κ =.24, recall=.44 vs. ICV κ =.70, recall=.86). The ICV detector is therefore quite good at selecting the correct class for a clip, and has much less bias towards labeling behavior as "not testing stated hypotheses".

Though not shown in Table 1, the ICV and RCF detectors were also compared to our original detectors [6], which used the original 73 features and had correlated features removed. Performance on the held-out test set was slightly worse than the RCF detector described here for designing controlled experiments (A'=.86, κ=.28, recall=.42), but slightly better for testing stated hypotheses (A'=.83, κ=.30, recall=.49). The new ICV detectors still outperform these detectors by a substantial amount. In sum, these findings support the idea that improving construct validity can lead to better overall prediction of systematic inquiry. Next, we determine if the ICV detectors can infer behavior with fewer actions.

4.2 Comparing Detectors' Performance Predicting with Less Data

The analyses here determine if detectors can predict behavior labeled at the clip level using less information. Again, these comparisons enable us to determine which detectors are more suitable for identifying which students need support *as they conduct their data collection*. Given our learning environment and approach, there are several ways to define "less information". We chose to look at simulation runs because they are the grain size at which we aim to activate scaffolding. In considering simulation runs, we also had to consider the clip number. Recall that several cycles of data collection could occur in an activity (each cycle represents a clip). Predictive performance could be impacted by the clip number under consideration, because later clips contain all actions associated with predecessor clips. Thus, we compare each detector on predicting behavior labeled at the clip level using actions up to the n^{th} run within the m^{th} clip, for varying numbers of runs and clips.

This approach required new sets of feature values to accommodate the fewer actions. Feature values were computed using all actions from clips 1..*m*-1 ($m > 1$), and all actions in the *m*th clip, up to and including the *n*th "sim start run" action (actions in dark grey in Figure 1). As an example, the feature values for the action sequence in Figure 1 for clip 2 and two runs would be computed using all actions 1-16 from the first clip, and actions up to and including the second "sim start run" (actions 31-38) in clip 2. Note that the notion of a "full run" actually spans several actions (e.g. actions 11-13 in Figure 1), given that the student could let the simulation run to completion, pause the simulation, or reset it. The "sim start run" action was chosen (rather than "sim finish" or "sim reset") to denote the boundary due to considerations for how we would scaffold students. In particular, we may want to prevent students from collecting of data unhelpful for the subsequent stage of inquiry, where they analyze data. Having the detectors classify behavior at the point where students try to run the simulation enables such an intervention.

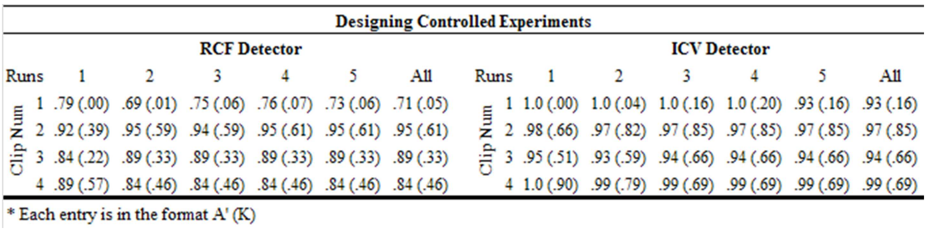
We compare detectors' performance using less data by comparing predictions for a given clip-run combination against the ground truth labels at the clip level. The number of clips was varied from 1 to 4, and the number of runs was varied from 1 to 5. A' and κ were computed per combination. Our expectation is that as the number of runs considered increases (and correspondingly the number of actions considered increases), A' and κ will increase. However, since many clips had fewer than five simulation runs, performance metrics may plateau as the number of runs increases. This may occur because no additional information would be available to improve predictions.

As shown in Table 2, the ICV detectors match or outperform the RCF detectors, when both detector variants are given less data on student performance. For clip 1, neither detector performed well for one or two runs (κ≅0.0). This finding associated with one run matched expectations because positive inquiry behavior can only be identified after two or more runs (cf. [7]). For runs 3-5 on the first clip, the RCF detector had A' ranging from .73 to .76, whereas the ICV detector had A' ranging from .93 to 1.0. The RCF detectors' κ remained at chance levels ranging from .06 to .07. The ICV detectors' κ values were better but still low, ranging from .16 to .20.

The designing controlled experiments detectors' poor performance on first clips may be due to misclassifications of such clips with exactly two runs (see Section 4.1). To see if ignoring such clips would impact detectors' ability to classify with less data, we removed them from the test set and re-computed our performance metrics. With only first clips with at least three runs, both detectors' performance using fewer actions, up to the first and second run, remained very low. However, when using actions up to runs 3-5, the ICV detector (run 3: A'=.99, κ=.42; run 4: A'=1.0, κ=.65; run 5: A'=.91, κ=.47) outperformed the RCF detector (A'=.70-.79, κ=.06-.11 for the same values). Additionally, three runs was the level at which the ICV detector could perform as well as classifying when considering all actions in the first clip (ICV all actions A'=.89, κ=.50).

For later clips within an activity, both detectors reach predictive performance equivalent to considering all actions (the "all" columns Table 2) after a single run.

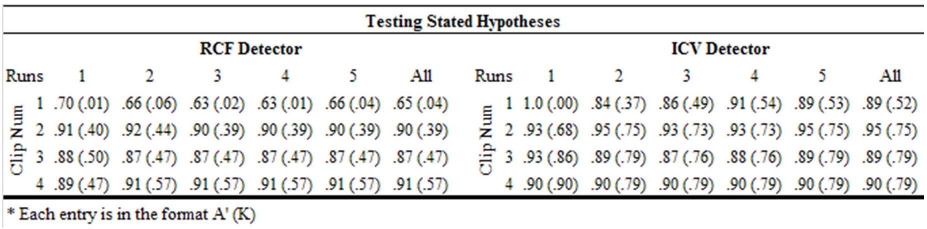
Table 2. Designing controlled experiments performance over *n*-runs and *m*-clips



Designing Controlled Experiments															
		RCF Detector								ICV Detector					
	Runs	1	2	3	4	5	All		Runs	1	2	3	4	5	All
Clip Num	1	.79 (.00)	.69 (.01)	.75 (.06)	.76 (.07)	.73 (.06)	.71 (.05)	Clip Num	1	1.0 (.00)	1.0 (.04)	1.0 (.16)	1.0 (.20)	.93 (.16)	.93 (.16)
	2	.92 (.39)	.95 (.59)	.94 (.59)	.95 (.61)	.95 (.61)	.95 (.61)		2	.98 (.66)	.97 (.82)	.97 (.85)	.97 (.85)	.97 (.85)	.97 (.85)
	3	.84 (.22)	.89 (.33)	.89 (.33)	.89 (.33)	.89 (.33)	.89 (.33)		3	.95 (.51)	.93 (.59)	.94 (.66)	.94 (.66)	.94 (.66)	.94 (.66)
	4	.89 (.57)	.84 (.46)	.84 (.46)	.84 (.46)	.84 (.46)	.84 (.46)		4	1.0 (.90)	.99 (.79)	.99 (.69)	.99 (.69)	.99 (.69)	.99 (.69)

* Each entry is in the format A' (K)

Table 3. Testing stated hypotheses performance over *n*-runs and *m*-clips



Testing Stated Hypotheses															
		RCF Detector								ICV Detector					
	Runs	1	2	3	4	5	All		Runs	1	2	3	4	5	All
Clip Num	1	.70 (.01)	.66 (.06)	.63 (.02)	.63 (.01)	.66 (.04)	.65 (.04)	Clip Num	1	1.0 (.00)	.84 (.37)	.86 (.49)	.91 (.54)	.89 (.53)	.89 (.52)
	2	.91 (.40)	.92 (.44)	.90 (.39)	.90 (.39)	.90 (.39)	.90 (.39)		2	.93 (.68)	.95 (.75)	.93 (.73)	.93 (.73)	.95 (.75)	.95 (.75)
	3	.88 (.50)	.87 (.47)	.87 (.47)	.87 (.47)	.87 (.47)	.87 (.47)		3	.93 (.86)	.89 (.79)	.87 (.76)	.88 (.76)	.89 (.79)	.89 (.79)
	4	.89 (.47)	.91 (.57)	.91 (.57)	.91 (.57)	.91 (.57)	.91 (.57)		4	.90 (.90)	.90 (.79)	.90 (.79)	.90 (.79)	.90 (.79)	.90 (.79)

* Each entry is in the format A' (K)

However, the ICV detectors outperform the RCF detectors. For example, when looking at clip 2 / run 2, the ICV detector performs better (A'=.97, κ=.82) than the RCF detector (A'=.95, κ=.59). Thus, once students have begun their second data collection cycle within an activity, the ICV detectors can better judge who needs scaffolding after the first run.

For testing stated hypotheses, the ICV detector again matched or outperformed the RCF detectors as shown in Table 3. For first clips, the RCF detector had A' values ranging from .63 to .70, and κ values at chance levels. However, the ICV detector performed well at this skill for first clips (ICV all actions A'=.89, κ=.52), a difference from designing controlled experiments. In fact, it could properly identify behavior after just the second run (ICV clip 1, run 2 had A'=.84, κ=.37). By the third run, predictive performance was on par with a detector that could consider all actions. For later clips, the ICV detector outperformed the RCF detector at all run levels. For example, when predicting using actions up to the second run for clip 2, the RCF detector had A'=.92 and κ=.44. Though this performance is good, the ICV detector performed much better with A'=.95 and κ=.75. Thus overall, the ICV detectors can be used to classify testing hypotheses behavior as early as the second run in the first clip, and are better at classification in later clips than the RCF detectors are.

5 Discussion and Conclusions

We investigated whether selecting features based on construct validity improves the predictive capabilities of machine-learned behavior detectors of scientific inquiry behaviors, designing controlled experiments and testing stated hypotheses, within a science microworld [10]. To explore this, we compared two types of detectors. One type removed used an automated approach, removing inter-correlated features (RCF detectors). Another used a partially manual approach to select features theoretically aligned with the behaviors, thereby increasing construct validity (ICV detectors). Models' predictive performance was compared against a held-out test set in two ways. We predicted behavior at the level of a full data collection cycle, the grain size at which behavior was labeled. We also predicted behavior at a finer grain size, microworld simulation runs, a grain size containing less information.

The results showed that improving construct validity can yield models with better overall predictive performance, even with less data. The ICV detector for testing stated hypotheses reached much higher performance levels than the RCF detector. The current ICV detector can effectively be used to trigger scaffolding when students finish data collection, given its high A'=.91 and κ=.70 values. It also can be used after as few as two runs on students' first data collection to provide fail-soft interventions that are not costly if misapplied. This is evidenced by A' values at or above .84, and κ at or above .37 found when increasing the number of simulation runs (thereby increasing the number of actions available) to make predictions.

The ICV detector for designing controlled experiments also outperformed its RCF counterpart. However, both the ICV and RCF detectors performed poorly when they inferred behavior for students' first data collection within an activity. We discovered this was due, in part, to poor classification of first cycles containing exactly two simulation runs. When ignoring such cycles, the ICV detector's performance improved substantially while the RCF detector remained poor. It could be applied in as few as three runs on students' first data collection. We believe the ICV detector failed on this case because the training set did not contain enough cases of this kind (see Section 2.3 for more details). This issue may be alleviated by adding more of these training clips and re-engineering the ICV detector following our procedure.

This paper offers two contributions towards leveraging feature-based machine-learned detectors to assess behavior. First, we explored the importance of considering construct validity when selecting features. We found that selecting features taking this into account yielded better detectors than selecting features using a more atheoretical approach, by removing inter-correlated features. Second, we described a general process for validating detectors at finer grain-sizes than they were trained and built. For our domain, the finer grain-size was the level of individual simulation runs. We found that detectors with improved construct validity could correctly infer behavior at the finer grain-size. This means we can reuse the ICV detectors as is to trigger scaffolding sooner, without needing to re-tag and retrain detectors to work at this level. In general, grain size and use of the detectors, whether for scaffolding (run or clip level in our domain) or for overall assessment (clip level in our domain), are both important to consider when evaluating detectors' applicability in a learning environment.

There are some limitations to this work. Though we controlled for the data mining algorithm and algorithm parameters, we did not compare the ICV detectors to others built using more sophisticated, automated feature selection approaches (e.g. [4], [5]). In addition, we only used a single data mining algorithm to generate detectors, J48 decision trees. Different data mining algorithms may have yielded different results. Our results are also contingent on the initial set features engineered, since there is no guarantee we computed all possible relevant features for our domain. Finally, we did not consider the notion of broader generalizability. For example, could a detector built for one science domain also detect inquiry skill in other domains? Considering these additional issues will provide more insight into the role construct validity plays in the development and successful use of machine-learned detectors.

Acknowledgements

This research is funded by the National Science Foundation (NSF-DRL#0733286, NSF-DRL#1008649, and NSF-DGE#0742503) and the U.S. Department of Education (R305A090170). Any opinions expressed are those of the authors and do not necessarily reflect those of the funding agencies.

References

1. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition. Morgan Kaufmann, San Francisco (2005)
2. Yu, L., Liu, H.: Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. In: Proc. of the 20th Int'l Conf. on Machine Learning, pp.856-863 (2003)
3. Pudil, P., Novovicova, J., Kittler, J.: Floating Search Methods in Feature Selection. Pattern Recognition Letters 15(11), 1119-1125 (1994)
4. Oh, I.-S., Lee, J.-S., Moon, B.-R.: Hybrid Genetic Algorithms for Feature Selection. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(11), 1424-1437 (Nov 2004)
5. Bernardini, A., Conati, C.: Discovering and Recognizing Student Interaction Patterns in Exploratory Learning Environments. In: Proc. of the 10th Int'l Conf. of Intelligent Tutoring Systems, pp.125-134 (2010)
6. Sao Pedro, M. A., Baker, R. S. J. d., Gobert, J. D., Montalvo, O., Nakama, A.: Leveraging Machine-Learned Detectors of Systematic Inquiry Behavior to Estimate and Predict Transfer of Inquiry Skill. User Modeling and User-Adapted Interaction (in press)
7. Chen, Z., Klahr, D.: All Other Things Being Equal: Acquisition and Transfer of the Control of Variables Strategy. Child Development 70(5), 1098-1120 (1999)
8. McElhaney, K., Linn, M.: Helping Students Make Controlled Experiments More Informative. In: Proc. of the 9th Int'l Conf. of the Learning Sciences, pp.786-793 (2010)
9. Buckley, B. C., Gobert, J., Horwitz, P.: Using Log Files to Track Students' Model-Based Inquiry. In: Proc. of the 7th Int'l Conf. of the Learning Sciences, pp.57-63 (2006)
10. Gobert, J., Sao Pedro, M., Baker, R., Toto, E., Montalvo, O.: Leveraging Educational Data Mining for Real Time Performance Assessment of Scientific Inquiry Skills within Microworlds. Journal of Educational Data Mining (accepted)
11. Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco, CA (1993)
12. Hanley, J. A., McNeil, B. J.: The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. Radiology 143, 29-36 (1982)
13. Ben-David, A.: About the Relationship between ROC Curves and Cohen's Kappa. Engineering Applications of Artificial Intelligence 21, 874-882 (2008)
14. Fogarty, J., Baker, R., & Hudson, S.: Case Studies in the Use of ROC Curve Analysis for Sensor-Based Estimates in Human Computer Interaction. In: Proc. of Graphics Interface, pp. 129-136 (2005).