Full Papers

Zone out no more: Mitigating mind wandering during computerized reading

Sidney K. D'Mello, Caitlin Mills, Robert Bixler, & Nigel Bosch
University of Notre Dame
118 Haggar Hall
Notre Dame, IN 46556, USA
sdmello@nd.edu

ABSTRACT

Mind wandering, defined as shifts in attention from task-related processing to task-unrelated thoughts, is a ubiquitous phenomenon that has a negative influence on performance and productivity in many contexts, including learning. We propose that next-generation learning technologies should have some mechanism to detect and respond to mind wandering in real-time. Towards this end, we developed a technology that automatically detects mind wandering from eye-gaze during learning from instructional texts. When mind wandering is detected, the technology intervenes by posing just-in-time questions and encouraging re-reading as needed. After multiple rounds of iterative refinement, we summatively compared the technology to a yoked-control in an experiment with 104 participants. The key dependent variable was performance on a post-reading comprehension assessment. Our results suggest that the technology was successful in correcting comprehension deficits attributed to mind wandering (d = .47 sigma) under specific conditions, thereby highlighting the potential to improve learning by "attending to attention."

Keywords

Mind wandering; gaze tracking; student modeling; attention-aware.

1. INTRODUCTION

Despite our best efforts to write a clear and engaging paper, chances are high that within the next 10 pages you might fall prey to what is referred to as zoning out, daydreaming, or mind wandering [45]. Despite your best intention to concentrate on our paper, at some point your attention might drift away to unrelated thoughts of lunch, childcare, or an upcoming trip. This prediction is not based on some negative or cynical opinion of the reader/reviewer (we read and review papers too), but on what is known about attentional control, vigilance, and concentration while individuals are engaged in complex comprehension activities, such as reading for understanding.

One recent study tracked mind wandering of 5,000 individuals from 83 countries with a smartphone app that prompted people with thought-probes at random intervals throughout the day [24]. People reported mind wandering for 46.9% of the prompts, which confirmed lab studies on the pervasiveness of mind wandering (see [45] for a review). Mind wandering is more than merely incidental; a recent meta-analysis of 88 samples indicated a negative correlation between mind wandering and performance across a variety of tasks [34], a correlation which increases with task complexity. When compounded with its high frequency, mind wandering can have serious consequences on the performance and productivity of society at large.

Mind wandering is also unfortunately an under-addressed problem in education and is yet to be deeply studied in the context of learning with technology. Traditional learning technologies rely on the assumption that students are attending to the learning session, although this is not always the case. For example, it has been estimated that students mind wander approximately 40% of the time when engaging with online lectures [38], which are an important component of MOOCs. Some advanced technologies do aim to detect and respond to affective states like boredom, but evidence for their effectiveness is still equivocal (see [9] for a review). Further, boredom is related to but not the same as attention [12]. There are technologies that aim to prevent mind wandering by engendering a highly immersive learning experience and have achieved some success in this regard [40, 41]. But what is to be done when attentional focus inevitably wanes as the session progresses and the novelty of the system and content fades?

Our central thesis is that next-generation learning technologies should include mechanisms to model and respond to learners' attention in real-time [8]. Such attention-aware technologies can model various aspects of learner attention (e.g., divided attention, alternating attention). Here, we focus on detecting and mitigating mind wandering, a quintessential signal of waning engagement. We situate our work in the context of reading because reading is a common activity shared across multiple learning technologies, thereby increasing the generalizability of our results. Further, students mind wander approximately 30% of the time during computerized reading [44]. And although mind wandering can facilitate certain cognitive processes like future planning and divergent thinking [2, 28], it negatively correlates with comprehension and learning (reviewed in [31, 45]), suggesting that it is important to address mind wandering during learning.

Towards this end, we developed and validated a closed-loop attention-aware learning technology that combines a machine-learned mind wandering detector with a real-time interpolated testing and re-study intervention. Our attention-aware technology works as follows. Learners read a text on a computer screen using a self-paced screen-by-screen (also called page-by-page) reading paradigm. We track eye-gaze during reading using a remote eye tracker that does not restrict head movements. We focus on eye-gaze for mind wandering detection due to decades of research suggesting a tight coupling between attentional focus and eye movements during reading [36]. When mind wandering is detected, the system intervenes in an attempt to redirect attentional focus and correct any comprehension deficits that might arise due to mind wandering. The interventions consist of asking comprehension question on pages where mind wandering was detected and providing opportunities to re-read based on learners' responses. In this paper, we discuss the mind wandering

detector, intervention approach, and results of a summative evaluation study[1].

1.1 Related Work

The idea of attention-aware user interfaces is not new, but was proposed almost a decade ago by Roda and Thomas [39]. There was even an article on futuristic applications of attention-aware systems in educational contexts [35]. Prior to this, Gluck, et al. [15] discussed the use of eye tracking to increase the bandwidth of information available to an intelligent tutoring system (ITS). Similarly, Anderson [1] followed up on some of these ideas by demonstrating how particular beneficial instructional strategies could only be launched via a real-time analysis of eye gaze.

Most of the recent work has been on leveraging eye gaze to increase the bandwidth of learner models [22, 23, 29]. Conati, et al. [5] provide an excellent review of much of the existing work in this area. We can group the research into three categories: (1) offline-analyses of eye gaze to study attentional processes, (2) computational modeling of attentional states, and (3) closed-loop systems that respond to attention in real-time. Offline-analysis of eye movements has received considerable attention in cognitive and educational psychology for several decades [e.g., 16, 19], so this area of research is relatively healthy. Online computational models of learner attention are just beginning to emerge [e.g., 6, 11], while closed-loop attention-aware systems are few and far between (see [7, 15, 42, 48] for a more or less exhaustive list). Two known examples, GazeTutor and AttentiveReview, are discussed below.

GazeTutor [7] is a learning technology for biology. It has an animated conversational agent that provides spoken explanations on biology topics which are synchronized with images. The system uses a Tobii T60 eye tracker to detect inattention, which is assumed to occur when learners' gaze is not on the tutor agent or image for at least five consecutive seconds. When this occurs, the system interrupts its speech mid utterance, directs learners to reorient their attention (e.g., "I'm over here you know"), and repeats speaking from the start of the current utterance. In an evaluation study, 48 learners (undergraduate students) completed a learning session on four biology topics with the attention-aware components enabled (experimental group) or disabled (control group). The results indicated that GazeTutor was successful in dynamically reorienting learners' attentional patterns towards the interface. Importantly, learning gains for deep reasoning questions were significantly higher for the experimental vs. control group, but only for high aptitude learners. The results suggest that even the most basic attention-aware technology can be effective in improving learning, at least for a subset of learners. However, a key limitation is that the researchers simply assumed that off-screen gaze corresponded to inattention, but did not test this assumption (e.g., students could have been concentrating with their eyes closed and this would have been perceived as being inattentive).

AttentiveReview [32] is a closed-loop system for MOOC learning on mobile phones. The system uses video-based photoplethysmography (PPG) to detect a learners' heart rate from the back camera of a smartphone while they view MOOC-like lectures on the phone. AttentiveReview ranks the lectures based on its estimates of learners' "perceived difficulty," selecting the most difficult lecture for subsequent review (called adaptive review). In a 32-participant between-subjects evaluation study, the authors found that learning gains obtained from the adaptive review condition were statistically on par with a full review condition, but were achieved in 66.7% less review time. Although this result suggests that AttentiveReview increased learning efficiency, there is the question as to whether the system should even be considered to be an "attention-aware" technology. This is because it is arguable if the system has anything to do with attention (except for "attention" appearing in its name) as it selects items for review based on a model of "perceived difficulty" and not on learners' "attentional state." The two might be related, but are clearly not the same.

1.2 Novelty

Our paper focuses on closing the loop between research on educational data and learning outcomes by developing and validating the first (in our view) real-time learning technology that detects and mitigates mind wandering during computerized reading. Although automated detection of complex mental states with the goal of developing intelligent learning technologies that respond to the sensed states is an active research area (see reviews by [9, 18]), mind wandering has rarely been explored as an aspect of a learner's mental state that warrants detection and corrective action. And while there has been some work on modeling the locus of learner attention (see review by [5]), mind wandering is inherently different than more commonly studied forms of attention (e.g., selective attention, distraction), because it involves more covert forms of involuntary attentional lapses spawned by self-generated internal thought [45]. Simply put, mind wandering is a form of "looking without seeing" because the eyes might be fixated on the appropriate external stimulus, but very little is being processed as the mind is consumed by stimulus-independent internal thoughts. *Offline* automated approaches to detect mind wandering have been developed (e.g., [3, 11, 27, 33]), but these detectors have not yet been used to trigger *online* interventions. Here, we adapt an offline gaze-based automated mind wandering detector [13] to trigger real-time interventions to address mind wandering during reading. We conduct a randomized control trial to evaluate the efficacy of our attention-aware learning technology in improving learning.

2. MIND WANDERING DETECTION

We adopted a supervised learning approach for mind wandering detection. Below we provide a high-level overview of the approach; readers are directed to [3, 13] for a detailed discussion of the general approach used to build gaze-based detectors of mind wandering.

2.1 Training Data

We obtained training data from a previous study [26] that involved 98 undergraduate students reading a 57-page text on the surface tension of liquids [4] on a computer screen for an average of 28 minutes. The text contained around 6500 words, with an average of 115 words per page, and was displayed on a computer screen with Courier New typeface. We recorded eye-gaze with a Tobii TX300 eye tracker set to a sampling frequency of 120 Hz.

[1] This paper reports updated results of an earlier version [10] presented as a "Late-Breaking Work" (LBW) poster at the *2016 ACM CHI conference*. LBW "Extended Abstracts" are not included in the main conference proceedings and copyright is retained by the authors.

Participants could read normally and were free to move or gesture as they pleased.

Participants were instructed to report mind wandering (during reading) by pressing a predetermined key when they found themselves "thinking about the task itself but *not the actual content of the text*" or when they were "thinking about *anything else besides the task*." This is consistent with contemporary approaches (see [45]) that rely on self-reporting because mind wandering is an internal conscious phenomena. Further, self-reports of mind wandering have been linked to predictable patterns in physiology [43], pupillometry [14], eye-gaze [37], and task performance [34], providing validity for this approach.

On average, we received mind wandering reports for 32% of the pages (*SD* = 20%), although there was considerable variability among participants (ranging from 0% to 82%). Self-reported mind wandering negatively correlated (r = -.23, p < .05) with scores on a subsequent comprehension assessment [26], which provides evidence for the predictive validity of the self-reports.

2.2 Model Building

The stream of eye-gaze data was filtered to produce a series of fixations, saccades, and blinks, from which *global eye gaze* features were extracted (see Figure 1). Global features are independent of the words being read and are therefore more generalizable than so-called local features. A full list of 62 global features along with detailed descriptions is provided in [13], but briefly the features can be grouped into the following four categories: (1) Eye movement descriptive features (n = 48) were statistical functionals (e.g., min, median) for fixation duration, saccade duration, saccade amplitude, saccade velocity, and relative and absolute saccade angle distributions; (2) Pupil diameter descriptive features were statistical functionals (n = 8) computed from participant-level z-score standardized estimates of pupil diameter; (3) Blink features (n = 2) consisted of the number of blinks and the mean blink duration; (4) Miscellaneous gaze features (n = 4) consisted of the number of saccades, horizontal saccade proportion, fixation dispersion, and the fixation duration/saccade duration ratio. We proceeded with a subset of 32 features after eliminating features exhibiting multicollinearity.

Features were calculated from only a certain amount of gaze data from each page, called the *window*. The end of the window was positioned 3 seconds before a self-report so as to not overlap with the key-press. The average amount of time between self-reports and the beginning of the page was 16 seconds. We used this time point as the end of the window for pages with no self-report. Pages that were shorter than the target window size were discarded, as were pages with windows that contained fewer than five gaze fixations as there was insufficient data to compute some of the features. There were a total of 4,225 windows with sufficient data for supervised classification.

We experimented with a number of supervised classifiers on window sizes of 4, 8, and 12 seconds to discriminate positive (pages with a self-report = 32%) from negative (pages without a self-report) instances of mind wandering. The training data were downsampled to achieve a 50% base rate; testing data were unaltered. A leave-one-participant-out validation approach was adopted where models were built on data from *n-1* participants and evaluated on the held-out participant. The process was repeated for all participants. Model validation was conducted in a way to simulate a real-time system by analyzing data from every page. When classification was not possible due to a lack of valid gaze data and/or because participants did not spend enough time on the page, we classified the page as a positive instance of mind wandering. This was done because analyses indicated that participants were more likely to be mind wandering in those cases (but see [13] for alternate strategies to handle missing instances).

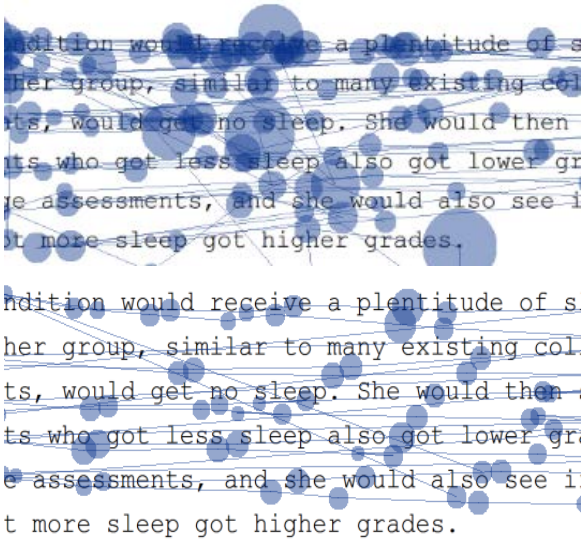


Figure 1: Gaze fixations during mind wandering (top) and normal reading (bottom)

2.3 Detector Accuracy

The best model was a support vector machine that used global features and operated on a window size of 8-seconds. The area under the ROC curve (AUC or AUROC or A') was .66, which exceeds the 0.5 chance threshold [17].

We assigned each instance as mind wandering or not mind wandering based on whether the detector's predicted likelihood of mind wandering (ranges from 0 to 1) was below or above 0.5 We adopted the default 0.5 threshold as it led to a higher rate of true positives while maintaining a moderate rate of true negatives. This resulted in the following confusion matrix shown in Table 1. The model had a weighted precision of 72.2% and a weighted recall of 67.4%, which we deemed to be sufficiently accurate for intervention.

Table 1: Proportionalized confusion matrix for mind wandering detection

	Predicted mind wandering (MW)	
Actual MW	*yes*	*no*
yes	0.715 (hit)	0.285 (miss)
no	0.346 (false positive)	0.654 (correct rejection)

3. Intervention to Address Mind Wandering

Our intervention approach is grounded in the basic idea that learning of conceptual information involves creating and maintaining an internal model (*mental model*) by integrating information from the text with prior knowledge from memory [25]. This integration process relies on attentional focus and breaks down during mind wandering because information from the external environment is no longer being integrated into the internal mental model. This results in an impaired model which leads to less effective suppression of off-task thoughts. This increase in mind wandering further impairs the mental model,

resulting in a vicious cycle. Our intervention targets this vicious cycle by redirecting attention to the primary task and attempting to correct for comprehension deficits attributed to mind wandering. Based on research demonstrating the effectiveness of interpolated testing [47], we propose that asking questions on pages where mind wandering is detected and encouraging re-reading in response to incorrect responses will aid in re-directing attention to the text and correct knowledge deficits.

3.1 Intervention Implementation

Our initial intervention was implemented for the same text used to create the mind wandering detector (although it could be applied to any text). The text was integrated into the computer reading interface. Mind wandering detection occurred when the learner navigated to the next page using the right arrow key. In order to address ambiguity in mind wandering detection, we used the detector's mind wandering likelihood to probabilistically determine when to intervene. For example, if the mind wandering likelihood was 70%, then there was a 70% chance of intervention on any given page (all else being equal). We did not intervene for the first three pages in order to allow the learner to become familiar with the text and interface. To reduce disruption, there was a 50% reduced probability of intervening on adjacent pages, and the maximum number of interventions was capped at 1/3 × the number of pages (19 for the present 57-page text). Table 2 presents pseudo code for when to launch an intervention.

Table 2: Pseudo code for intervention strategy

```
launch_intervention:
    if current_page >= WAITPAGES
    and
        total_interventions < MAXINTRV)
    and
        gaze_likelihood > random(0,1)
    and
        (!has_intervened(previous_page)
        or 0.5 < random (0,1)):
            do_intervention()
    else:
        show_next_page()

do_intervention:
    answer1 = show_question1()
    if answer1 is correct:
        show_positive_feedback()
        show_next_page()
    else:
        show_neg_feedback()
        suggest_rereading()
        if page advance detected:
            answer2 = show_question2();
            show_next_page()
```

Figure 2 presents an outline of the intervention strategy. The intervention itself relied on two multiple choice questions for each page (screen) of the text. When the system decided to intervene, one of the questions (randomly selected) was presented to the learner. If the learner answered this *online question* correctly, positive feedback was provided, and the learner could advance to the next page. If the learner answered incorrectly, negative feedback was provided, and the system encouraged the learner to re-read the page. The learner was then provided with a second (randomly selected) online question, which could either be the same or the alternate question for that page. Feedback was not provided and the learner was allowed to advance to the next page regardless of whether the second question was answered correctly, so as not to be overly burdensome.

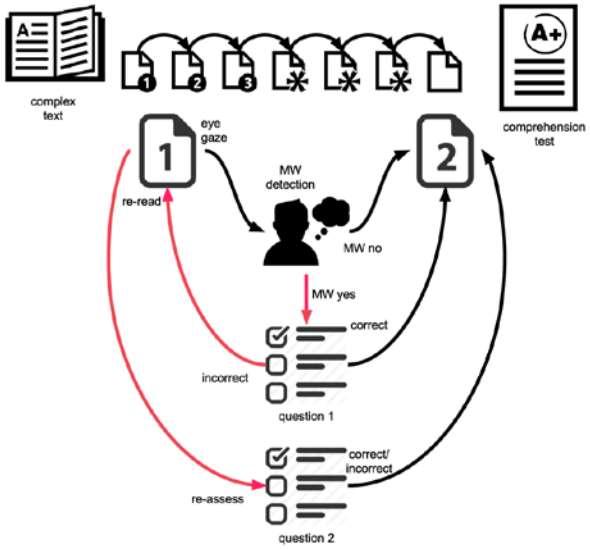


Figure 2: Outline of intervention strategy

3.2 Iterative Refinement

The technology was refined through multiple rounds of formative testing with 67 participants, recruited from the same institution used to build the detector. Participants were observed while interacting with the technology, their responses were analyzed, and they were interviewed about their experience. We used the feedback gleaned from these tests to refine the intervention parameters (i.e., when to launch, how many interventions to launch, whether to launch interventions on subsequent pages), intervention questions themselves, and instructions on how to attend to the intervention. For example, earlier versions of the intervention used a fixed threshold (instead of the aforementioned probabilistic approach) to trigger an intervention. Despite many attempts to set this threshold, the end result was that some participants received many interventions while others received almost no interventions. This issue was corrected by probabilistically rather than deterministically launching the intervention. Additional testing/refinement of the comprehension questions used in the intervention was done using crowdsourcing platforms, specifically Amazon's Mechanical Turk (MTurk).

4. Evaluation Study

We conducted a randomized controlled trial to evaluate the technology. The experiment had two conditions: an intervention condition and a yoked control condition (as described below). The yoked control was needed to verify that any learning benefits are attributed to the technology being sensitive to mind wandering and not merely to the added opportunities to answer online questions and re-read. This is because we know that interpolated testing itself has beneficial comprehension effects [47].

4.1 Method

Participants (N = 104) were a new set of undergraduate students who participated to fulfill research credit requirements. They were recruited from the same university used to build the MW detector and for the iterative testing and refinement cycles.

We did not use a pretest because we expected participants to be unfamiliar with the topic. Participants were not informed that the interface would be tracking their mind wandering (until the

debriefing at the end), Instead, they were instructed as follows: "While reading the text, you will occasionally be asked some questions about the page you just read. Depending on your answer, you will re-read the same page and you will be asked another question that may or may not be the same question."

Participants in the intervention condition received the intervention as described above (i.e., based on detected mind wandering likelihoods). Each participant in the yoked control condition was *paired* with a participant in the intervention condition. He or she received an intervention question on the same pages as their paired intervention participant regardless of mind wandering likelihood. For example, if participant A (i.e., intervention condition) received questions on pages 5, 7, 10, and 25, participant B (i.e., yoked control condition) would receive intervention questions on the same pages. However, if the yoked participant answered incorrectly, then (s)he had the opportunity to re-read and answer another question regardless of the outcome of their intervention-condition partner.

After reading, participants completed a 38-item multiple choice comprehension assessment to measure learning. The questions were randomly selected from the 57 pages (one per page) with the exception that a higher selection priority was given to pages that were re-read on account of the intervention. Participants in the yoked control condition received the same posttest questions as their intervention condition counterparts.

4.2 Results

Participants received an average of 16 (min of 7 and max of 19) interventions. They spent an average of 27.5 seconds on each screen prior to receiving an intervention. There was no significant difference across conditions (p = .998), suggesting that reading time was not a confound. In what follows, we compared each intervention participant to his/her yoked control with a two-tailed paired-samples t-test and a 0.05 criteria for statistical significance.

Mind wandering detection. The detector's likelihood of mind wandering was slightly higher for participants in the yoked-control condition (M = .431; SD = .170) compared to the intervention condition (M = .404; SD = .112), but the difference was not statistically significant (p = .348). This was unsurprising as participants in both groups received the same interventions, which itself was expected to reduce mind wandering. Importantly, mind wandering likelihoods were negatively correlated with performance on the online questions (r = -.296, p = .033) as well as on posttest questions (r = -.319, p = .021). This provides evidence for the validity of the mind wandering detector when applied to a new set of learners and under different conditions (i.e., reading interspersed with online questions compared to uninterrupted reading).

Comprehension assessment. There was some overlap between the online questions and the posttest questions. To obtain an unbiased estimate of learning, we only analyzed performance on previously unseen posttest questions. That is, questions that were used as part of the intervention were first removed before computing posttest scores.

There were no significant condition differences on overall posttest scores (p = .846). The intervention condition answered 57.6% (SD = .157) of the questions correctly while the yoked control condition answered 58.1% (SD = .129) correctly. This finding was not surprising as both conditions received the exact same treatment except that the interventions were triggered based on detected mind wandering in the intervention condition but not the control condition.

Next, we examined posttest performance as a function of mind wandering during reading. Each page was designated as a low or high mind wandering page based on a median split of mind wandering likelihoods (medians = .35 and .36 on a 0 to 1 scale for intervention and control conditions, respectively). We then analyzed performance on posttest questions corresponding to pages with low vs. high likelihoods of mind wandering (during reading). The results are shown in Table 3.

We found no significant posttest differences on pages where both the intervention and control participants had low (p = .759) or high (p = .922) mind wandering likelihoods (first and last rows in Table 3, respectively). There was also no significant posttest difference (p = .630) for pages where the intervention condition had high mind wandering likelihoods but the control condition had low mind wandering likelihoods (row 3). However, the intervention condition significantly (p = .003, d = .47 sigma) outperformed the control condition for pages where the intervention participants had low likelihoods of mind wandering but control participants had high mind wandering likelihoods (row 2). These last two finding suggests that the intervention had the intended effect of reducing comprehension deficits attributable to mind wandering because it led to equitable performance when mind wandering was high and improved performance when it was low.

Table 3: Posttest performance (proportion of correct responses) as a function of mind wandering during reading. Standard deviations in parenthesis.

	Mind wandering		**Posttest scores**	
N	*Int.*	*Cntrl.*	*Int.*	*Cntrl.*
43	Low	Low	.604 (.288)	.623 (.287)
40	**Low**	**High**	**.643 (.263)**	**.489 (.298)**
43	High	Low	.535 (.295)	.566 (.305)
45	High	High	.522 (.312)	.515 (.291)

Note. Int. = intervention. Cntrl. = control. Bolded cells represent a statistically significant difference. N = number of pairs (out of 52) in each analysis. It differs slightly across analyses as not all participants were assigned to each mind wandering group.

After-task interview. We interviewed a subset of the participants in order to gauge their subjective experience with the intervention. A few key themes emerged. Participants reported paying closer attention to the text after realizing they would be periodically answering multiple-choice questions. This was good. However, participants also reported that they adapted their reading strategies in one of two ways in response to the questions. Since the questions targeted factual information (sometimes verbatim) from the text, some participants paid more attention to details and precise wordings instead of the broader concepts being discussed in the text. More discouragingly, some participants reported adopting a preemptive skimming strategy in that they would only look for keywords that they expected to appear in a subsequent question.

Participants were encouraged to re-read text when they answered incorrectly before receiving another question (or the same question in some cases). Many participants reported simply scanning the text (when re-reading) to locate keywords from the question before moving on. Since the scanning strategy was often

successful to answer the subsequent question, participants reported that the questions were too easy and it took relatively little effort to locate the correct answer compared to re-reading. They suggested that it may have been better if the questions had targeted key concepts rather than facts.

Finally, participants reported difficulties with re-engaging with the text after answering an online question because the text was cleared when an intervention question was displayed; an item that can be easily corrected in subsequent versions.

5. Discussion

We developed the first educational technology capable of real-time mind wandering detection and dynamic intervention during computerized reading. In the remainder of this section, we discuss the significance of our main findings, limitations, and avenues for future work.

5.1 Significance of Main Findings

We have three main findings. First, we demonstrated that a machine-learned mind wandering detector built in one context can be applied to a different (albeit related) interaction context. Specifically, the detector was trained on a data set involving participants silently reading and self-reporting mind wandering, but was applied to an interactive context involving interpolated assessments, which engendered different reading strategies. Further, self-reports of mind wandering were *not* collected in this interactive context, which might have influenced mind wandering rates in and of itself. Despite these differences, we were able to demonstrate the predictive validity of the detector by showing that it negatively correlated with both online and offline comprehension scores when evaluated on new participants.

Second, we showed promising effects for our intervention approach despite a very conservative experimental design, which ensured that the intervention and control groups were equated along all respects, except that the intervention was triggered based on the mind wandering detector (key manipulation). Further, we used a probabilistic approach to trigger an intervention, because the detector is inherently imperfect. As a result, participants could have received an intervention when they were not mind wandering and/or could have failed to receive one when they were mind wandering. Therefore, it was essential to compare the two groups under conditions when the mind wandering levels differed. This more nuanced analysis revealed that although the intervention itself did not lead to a boost in overall comprehension (because it is remedial), it equated comprehension scores when mind wandering was high (i.e., scores for the intervention group were comparable when the control group was low on mind wandering). It also demonstrated the cost of not intervening during mind wandering (i.e., scores for the intervention group were greater when the control group was high on mind wandering). In other words, the intervention was successful in mitigating the negative effects of mind wandering.

Third, despite the advantages articulated above, the intervention itself was reactive and engendered several unintended (and presumably suboptimal) behaviors. In particular, students altered their reading strategies in response to the interpolated questions, which were a critical part of the intervention. In a sense, they attempted to "game the intervention" by attempting to proactively predict the types of questions they might receive and then adopting a complementary reading strategy consisting of skimming and/or focusing on factual information. This reliance on surface- rather than deeper-levels of processing was incongruent with our goal of promoting deep comprehension.

5.2 Limitations

There are a number of methodological limitations with this work that go beyond limitations with the intervention (as discussed above). First, we focused on a single text that is perceived as being quite dull and consequently triggers rather high levels of mind wandering [26]. This raises the question of whether the detector will generalize to different texts. We expect some level of generalizability in terms of features used because the detector only used content- and position- (on the screen) free global gaze features. However, given that several supervised classifiers are very sensitive to differences in base rates, the detector might over- or under- predict mind wandering when applied to texts that engender different rates of mind wandering. Therefore, retraining the detector with a more diverse set of texts is warranted.

Another limitation is the scalability of our learning technology. The eye tracker we used was a cost-prohibitive Tobii TX300 that will not scale beyond the laboratory. Fortunately, commercial-off-the-shelf (COTS) eye trackers, such as Eye Tribe and Tobii EyeX, can be used to surpass this limitation. It is an open question as to whether the mind wandering detector can operate with similar fidelity with these COTS eye trackers. Our use of global gaze features which do not require high-precision eye tracking holds considerable promise in this regard. Nevertheless, replication with scalable eye trackers and/or scalable alternatives to eye tracking (e.g., facial-feature tracking [46] or monitoring reading patterns [27]) is an important next step (see Section 5.3).

Our use of surface-level questions for both the intervention and the subsequent comprehension assessment is also a limitation as is the lack of a delayed comprehension assessment. It might be the case that the intervention effects manifest as richer encodings in long-term memory, a possibility that cannot be addressed in the current experiment that only assessed immediate learning.

Other limitations include a limited student sample (i.e. undergraduates from a private Midwestern college) and a laboratory setup. It is possible that the results would not generalize to a more diverse student population or in more ecological environments (but see below for evidence of generalizability of the detector in classroom environments). Replication with data from more diverse populations and environments would be a necessary next step to increase the ecological validity of this work.

5.3 Future Work

Our future work is progressing along two main fronts. One is to address limitations in the intervention and design of the experimental evaluation as discussed above. Accordingly, we are exploring alternative intervention strategies, such as: (a) tagging items for future re-study rather than interrupting participants during reading; (b) highlighting specific portions of the text as an overt cue to facilitate comprehension of critical information; (c) asking fewer intervention questions, but selecting inference questions that target deeper levels of comprehension and that span multiple pages of the text; and (d) asking learners to engage in reflection by providing written self-explanations of the textual content. We are currently evaluating one such redesigned intervention – open-ended questions targeting deeper levels of comprehension (item c). Our revised experimental design taps both surface- and inference-level comprehension and assesses comprehension immediately after reading (to measure learning) and after a one-week delay (to measure retention).

We are also developing attention-aware versions of more interactive interfaces, such as learning with an intelligent tutoring

system called GuruTutor [30]. This project also addresses some of the scalability concerns by replacing expensive research-grade eye tracking with cost-effective COTS eye tracking (e.g., the Eye Tribe or Tobii EyeX) and provides evidence for real-world generalizability by collecting data in classrooms rather than the lab. We recently tested our implementation on 135 students (total) in a noisy computer-enabled high-school classroom where eye-gaze of entire classes of students was collected during their normal class periods [20]. Using a similar approach to the present work, we used the data to build and validate a student-independent gaze-based mind wandering detector. The resultant mind wandering detection accuracy (F_1 of 0.59) was substantially greater than chance (F_1 of 0.24) and outperformed earlier work on the same domain [21]. The next step is to develop interventions that redirect attention and correct learning deficiencies attributable to mind wandering and to test the interventions in real-world environments. By doing so, we hope to advance our foundational vision of developing next-generation technologies that enhance the process and products of learning by "attending to attention."

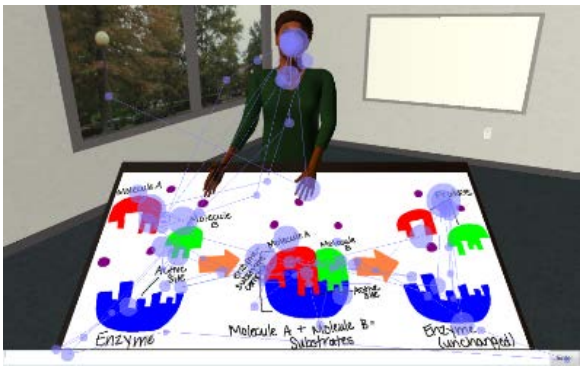


Figure 3: Guru Tutor interface overlaid with eye-gaze obtained via the EyeTribe

6. Acknowledgements

This research was supported by the National Science Foundation (NSF) (DRL 1235958 and IIS 1523091). The authors are grateful to Kris Kopp and Jenny Wu for their contributions to the study. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

7. REFERENCES

[1] Anderson, J.R. 2002. Spanning seven orders of magnitude: A challenge for cognitive modeling. *Cognitive Science*, *26* (1), 85-112.

[2] Baird, B., Smallwood, J., Mrazek, M.D., Kam, J.W., Franklin, M.S. and Schooler, J.W. 2012. Inspired by distraction mind wandering facilitates creative incubation. *Psychological Science*, *23* (10), 1117-1122.

[3] Bixler, R. and D'Mello, S.K. 2016. Automatic gaze-based user-independent detection of mind wandering during computerized reading. *User Modeling & User-Adapted Interaction*, *26*, 33-68.

[4] Boys, C.V. 1895. *Soap bubbles, their colours and the forces which mold them*. Society for Promoting Christian Knowledge.

[5] Conati, C., Aleven, V. and Mitrovic, A. 2013. Eye-Tracking for Student Modelling in Intelligent Tutoring Systems. In Sottilare, R., Graesser, A., Hu, X. and Holden, H. eds. *Design Recommendations for Intelligent Tutoring Systems - Volume 1: Learner Modeling*, Army Research Laboratory, Orlando, FL.

[6] Conati, C. and Merten, C. 2007. Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Knowledge-Based Systems*, *20* (6), 557-574.

[7] D'Mello, S., Olney, A., Williams, C. and Hays, P. 2012. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of human-computer studies*, *70* (5), 377-398.

[8] D'Mello, S.K. 2016. Giving Eyesight to the Blind: Towards attention-aware AIED. *International Journal of Artificial Intelligence In Education*, *26* (2), 645-659.

[9] D'Mello, S.K., Blanchard, N., Baker, R., Ocumpaugh, J. and Brawner, K. 2014. I feel your pain: A selective review of affect-sensitive instructional strategies. In Sottilare, R., Graesser, A., Hu, X. and Goldberg, B. eds. *Design Recommendations for Adaptive Intelligent Tutoring Systems: Adaptive Instructional Strategies (Volume 2)*, US Army Research Laboratory, Orlando, FL.

[10] D'Mello, S.K., Kopp, K., Bixler, R. and Bosch, N. 2016. Attending to attention: Detecting and combating mind wandering during computerized reading In *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2016)*, ACM, New York.

[11] Drummond, J. and Litman, D. 2010. In the zone: Towards Detecting student zoning out using supervised machine learning. In Aleven, V., Kay, J. and Mostow, J. eds. *Intelligent Tutoring Systems.*, Springer-Verlag, Berlin / Heidelberg.

[12] Eastwood, J.D., Frischen, A., Fenske, M.J. and Smilek, D. 2012. The unengaged mind: Defining boredom in terms of attention. *Perspectives on Psychological Science*, *7* (5), 482-495.

[13] Faber, M., Bixler, R. and D'Mello, S.K. in press. An automated behavioral measure of mind wandering during computerized reading. *Behavior Research Methods*.

[14] Franklin, M.S., Broadway, J.M., Mrazek, M.D., Smallwood, J. and Schooler, J.W. 2013. Window to the Wandering Mind: Pupillometry of Spontaneous Thought While Reading. *The Quarterly Journal of Experimental Psychology*, *66* (12), 2289-2294.

[15] Gluck, K.A., Anderson, J.R. and Douglass, S.A. 2000. Broader Bandwidth in Student Modeling: What if ITS Were "Eye" TS? In Gauthier, C., Frasson, C. and VanLehn, K. eds. *Proceedings of the 5th international conference on intelligent tutoring systems*, Springer, Berlin.

[16] Graesser, A., Lu, S., Olde, B., Cooper-Pye, E. and Whitten, S. 2005. Question asking and eye tracking during cognitive disequilibrium: Comprehending illustrated texts on devices when the devices break down. *Memory and Cognition*, *33*, 1235-1247.

[17] Hanley, J.A. and McNeil, B.J. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143* (1), 29-36.

[18] Harley, J.M., Lajoie, S.P., Frasson, C. and Hall, N.C. in press. Developing Emotion-Aware, Advanced Learning Technologies: A Taxonomy of Approaches and Features. *International Journal of Artificial Intelligence In Education*.

[19] Hegarty, M. and Just, M. 1993. Constructing mental models of machines from text and diagrams. *Journal of Memory and Language*, *32* (6), 717-742.

[20] Hutt, S., Mills, C., Bosch, N., Krasich, K., Brockmole, J.R. and D'Mello, S.K. in review. Out of the Fr-Eye- ing Pan: Towards Gaze-Based Models of Attention during Learning with Technology in the Classroom.

[21] Hutt, S., Mills, C., White, S., Donnelly, P.J. and D'Mello, S.K. 2016. The Eyes Have It: Gaze-based Detection of Mind Wandering during Learning with an Intelligent Tutoring System. In *Proceedings of the 9th International Conference on Educational Data Mining (EDM 2016)*, International Educational Data Mining Society.

[22] Jaques, N., Conati, C., Harley, J.M. and Azevedo, R. Year. Predicting Affect from Gaze Data during Interaction with an Intelligent Tutoring System. In *Intelligent Tutoring Systems*, (2014), Springer, 29-38.

[23] Kardan, S. and Conati, C. 2012. Exploring gaze data for determining user learning with an interactive simulation. In Carberry, S., Weibelzahl, S., Micarelli, A. and Semeraro, G. eds. *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization (UMAP 2012)*, Springer, Berlin.

[24] Killingsworth, M.A. and Gilbert, D.T. 2010. A wandering mind is an unhappy mind. *Science*, *330* (6006), 932-932.

[25] Kintsch, W. 1998. *Comprehension: A paradigm for cognition*. Cambridge University Press, New York.

[26] Kopp, K., D'Mello, S. and Mills, C. 2015. Influencing the occurrence of mind wandering while reading. *Consciousness and Cognition*, *34* (1), 52-62.

[27] Mills, C. and D'Mello, S.K. 2015. Toward a Real-time (Day) Dreamcatcher: Detecting Mind Wandering Episodes During Online Reading. In Romero, C., Pechenizkiy, M., Boticario, J. and Santos, O. eds. *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, International Educational Data Mining Society.

[28] Mooneyham, B.W. and Schooler, J.W. 2013. The costs and benefits of mind-wandering: A review. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, *67* (1), 11.

[29] Muir, M. and Conati, C. 2012. An analysis of attention to student–adaptive hints in an educational game. In Cerri, S.A., Clancey, W.J., Papadourakis, G. and Panourgia, K. eds. *Proceedings of the International Conference on Intelligent Tutoring Systems*, Springer, Berlin.

[30] Olney, A., D'Mello, A., Person, N., Cade, W., Hays, P., Williams, C., Lehman, B. and Graesser, A. 2012. Guru: A computer tutor that models expert human tutors. In Cerri, S., Clancey, W., Papadourakis, G. and Panourgia, K. eds. *Proceedings of the 11th International Conference on Intelligent Tutoring Systems*, Springer-Verlag, Berlin/Heidelberg.

[31] Olney, A., Risko, E.F., D'Mello, S.K. and Graesser, A.C. 2015. Attention in educational contexts: The role of the learning task in guiding attention. In Fawcett, J., Risko, E.F. and Kingstone, A. eds. *The Handbook of Attention*, MIT Press, Cambridge, MA.

[32] Pham, P. and Wang, J. 2016. Adaptive Review for Mobile MOOC Learning via Implicit Physiological Signal Sensing. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction (ICMI 2016)*, ACM, New York, NY.

[33] Pham, P. and Wang, J. 2015. AttentiveLearner: improving mobile MOOC learning via implicit heart rate tracking. In *International Conference on Artificial Intelligence in Education*, Springer, Berlin Heidelberg.

[34] Randall, J.G., Oswald, F.L. and Beier, M.E. 2014. Mind-wandering, cognition, and performance: A theory-driven meta-analysis of attention regulation. *Psychological Bulletin*, *140* (6), 1411-1431.

[35] Rapp, D.N. 2006. The value of attention aware systems in educational settings. *Computers in Human behavior*, *22* (4), 603-614.

[36] Rayner, K. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, *124* (3), 372-422.

[37] Reichle, E.D., Reineberg, A.E. and Schooler, J.W. 2010. Eye movements during mindless reading. *Psychological Science*, *21* (9), 1300.

[38] Risko, E.F., Buchanan, D., Medimorec, S. and Kingstone, A. 2013. Everyday attention: mind wandering and computer use during lectures. *Computers & Education*, *68* (1), 275-283.

[39] Roda, C. and Thomas, J. 2006. Attention aware systems: Theories, applications, and research agenda. *Computers in Human Behavior*, *22* (4), 557-587.

[40] Rowe, J., Mott, B., McQuiggan, S., Robison, J., Lee, S. and Lester, J. Year. Crystal island: A narrative-centered learning environment for eighth grade microbiology. In *Workshop on Intelligent Educational Games at the 14th International Conference on Artificial Intelligence in Education, Brighton, UK*, (2009), 11-20.

[41] Shute, V.J., Ventura, M., Bauer, M. and Zapata-Rivera, D. 2009. Melding the power of serious games and embedded assessment to monitor and foster learning: Flow and grow. In Ritterfeld, U., Cody, M. and Vorderer, P. eds. *Serious games: Mechanisms and effects*, Routledge, Taylor and Francis, Mahwah, NJ.

[42] Sibert, J.L., Gokturk, M. and Lavine, R.A. 2000. The reading assistant: eye gaze triggered auditory prompting for reading remediation. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, ACM, New York, NY.

[43] Smallwood, J., Davies, J.B., Heim, D., Finnigan, F., Sudberry, M., O'Connor, R. and Obonsawin, M. 2004. Subjective experience and the attentional lapse: Task engagement and disengagement during sustained attention. *Consciousness and Cognition*, *13* (4), 657-690.

[44] Smallwood, J., Fishman, D.J. and Schooler, J.W. 2007. Counting the cost of an absent mind: Mind wandering as an underrecognized influence on educational performance. *Psychonomic Bulletin & Review*, *14* (2), 230-236.

[45] Smallwood, J. and Schooler, J.W. 2015. The science of mind wandering: empirically navigating the stream of consciousness. *Annu. Rev. Psychol*, *66*, 487-518.

[46] Stewart, A., Bosch, P., Chen, H., Donnelly, P.J. and D'Mello, S.K. 2016. Where's Your Mind At? Video-Based Mind Wandering Detection During Film Viewing. In Aroyo, L., D'Mello, S., Vassileva, J. and Blustein, J. eds. *Proceedings of the 2016 ACM on International Conference on User Modeling, Adaptation, & Personalization (ACM UMAP 2016)*, ACM, New York.

[47] Szpunar, K.K., Khan, N.Y. and Schacter, D.L. 2013. Interpolated memory tests reduce mind wandering and improve learning of online lectures. *Proceedings of the National Academy of Sciences*, *110* (16), 6313-6317.

[48] Wang, H., Chignell, M. and Ishizuka, M. 2006. Empathic tutoring software agents using real-time eye tracking. In *Proceedings of the 2006 symposium on Eye tracking research &applications*, ACM, New York.

Measuring Similarity of Educational Items Using Data on Learners' Performance

Jiří Řihák
Faculty of Informatics
Masaryk University
Brno, Czech Republic
thran@mail.muni.cz

Radek Pelánek
Faculty of Informatics
Masaryk University
Brno, Czech Republic
pelanek@mail.muni.cz

ABSTRACT

Educational systems typically contain a large pool of items (questions, problems). Using data mining techniques we can group these items into knowledge components, detect duplicated items and outliers, and identify missing items. To these ends, it is useful to analyze item similarities, which can be used as input to clustering or visualization techniques. We describe and evaluate different measures of item similarity that are based only on learners' performance data, which makes them widely applicable. We provide evaluation using both simulated data and real data from several educational systems. The results show that Pearson correlation is a suitable similarity measure and that response times are useful for improving stability of similarity measures when the scope of available data is small.

1. INTRODUCTION

Interactive educational systems offer learners items (problems, questions) for solving. Realistic educational systems typically contain a large number of such items. This is particularly true for adaptive systems, which try to present suitable items for different kinds of learners. The management of a large pool of items is difficult. However, educational systems collect data about learners' performance and the data can be used to get insight into item properties. In this work we focus on methods for computing item similarities based on learners' performance data, which consists of binary information about the answers (correct/incorrect).

Automatically detected item similarities are the first and necessary step in further analysis such as clustering of the items, which is useful in several ways, with one particular application being learner modeling [9]. Learner models estimate knowledge and skills of learners and are the basis of adaptive behavior of educational systems. A learner's models requires a mapping of items into knowledge components [17]. Item clusters can serve as a basis for knowledge component definition or refinement. The specified knowledge components are relevant not only for modeling, but they are typically directly visible to learners in the user interface of a system, e.g., in a form of open learner model visualizing the estimated knowledge state, or in a personalized overview of mistakes, which is grouped by knowledge components.

Information about items is also very useful for management of the content of educational systems – preparation of new items, filtering of unsuitable items, preparation of explanations, and hint messages. Information about item similarities and clusters can be also relevant for teachers as it can provide them an inspiration for "live" discussions in class. This type of applications is in line with Baker's argument [1] for focusing on the use of learning analytics for "leveraging human intelligence" instead of its use for automatic intelligent methods.

Item similarities and clusters are studied not only in educational data mining but also in a closely related area of recommender systems. The setting of recommender systems is in many aspects very similar to educational systems – in both cases we have users and items, just instead of "performance" (the correctness of answers, the speed of answers) recommender systems consider "ratings" (how much a user likes an item). Item similarities and clustering techniques have thus been also considered in the recommender systems research (we mention specific techniques below). There is a slight, but important difference between the two areas. In recommender systems item similarities and clusterings are typically only auxiliary techniques hidden within a "recommendation black box". In educational system, it is useful to make these results explicitly available to system developers, curriculum production teams, or teachers.

There are two basic approaches to dealing with item similarities and knowledge components: a "model based approach" and an "item similarity approach". The basic idea of the model based approach is to construct a simplified model that explains the observed data. Based on a matrix of learners' answers to items we construct a model that predicts these answers. Typically, the model assigns several latent skills to learners and uses a mapping of items to corresponding latent factors. This kind of models can often be naturally expressed using matrix multiplication, i.e., fitting a model leads to matrix factorization. Once we fit the model to data, items that have the same value of a latent factor can be denoted as "similar". This approach leads naturally to multiple knowledge components per skill. The model is typically computed

using some optimization technique that leads only to local optima (e.g., gradient descent). It is thus necessary to address the role of initialization, and parameter setting of the search procedure. In recommender systems this approach is used for implementation of collaborative filtering; it is often called "singular value decomposition" (SVD) [18]. In educational context many variants of this approach have been proposed under different names and terminology, e.g., Q-matrix [3], non-negative matrix factorization techniques [8], sparse factor analysis [19], or matrix refinement [10].

With the item similarity approach we do not construct an explicit model of learners' behavior, but we compute directly a similarity measure for each pairs of items. These similarities are then used to compute clusters of items, to project items into a plane, or for other analysis (e.g., for each item listing the 3 most similar items). This approach naturally leads to a mapping with a single knowledge component per item (i.e., different kind of output from most model based methods). One advantage of this approach is easier interpretability. In recommender system research this approach is called neighborhood-based methods [11] or item-item collaborative filtering [7]. Similarity has been used for clustering of items [23, 24] and also for clustering of users [29]. In educational setting item similarity has been analyzed using correlation of learners' answers [22] and problem solving times [21], and also using learners' wrong answers [25].

So far we have discussed methods that are based only on data about learners' answers. Often we have some additional information about items and their similarities, e.g., a manual labeling or data based on syntactic similarity of items (text of questions). For both model based and item similarity approaches previous research has studied techniques for combination of these different types of inputs [10, 21].

In this work we focus on the item similarity approach, because in the educational setting this approach is less explored than the model based approach. We discuss specific techniques, clarify details of their usage, and provide evaluation using both data from real learners and simulated data. Simulated data are useful for evaluation of the considered unsupervised machine learning tasks, because in the case of real-world data we do not know the "ground truth".

The specific contributions of this work are the following. We provide guidelines for the choice of item similarity measures – we discuss different options and provide results identifying suitable measures (Pearson, Yule, Cohen); we also demonstrate the usefulness of "two step similarity measures". We explore benefits of the use of response time information as supplement to usual information of correctness of answer. We use and discuss several evaluation methods for the considered tasks. We specifically consider the issue of "how much data do we need". This is often practically more important than the exact choice of a used technique, but the issue is rather neglected in previous work.

2. MEASURES OF ITEM SIMILARITY

Figure 1 provides a high-level illustration of the item similarity approach. This approach consist of two steps that are to a large degree independent. At first, we compute an item similarity matrix, i.e., for each pair of items i, j we compute similarity s_{ij} of these items. At second, we can construct clusters or visualizations of items using only the item similarity matrix.

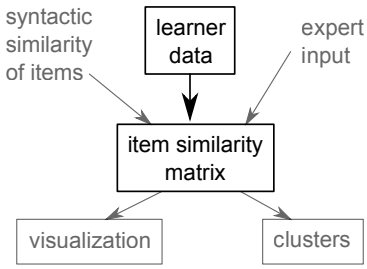


Figure 1: High-level illustration of the general approach to item analysis based on item similarities.

Experience with clustering algorithms suggests that the appropriate choice of similarity measure is more important than choice of clustering algorithm [13]. The choice of similarity measure is domain specific and it is typically not explored in general research on clustering. Therefore, we focus on the first step – the choice of similarity measure – and explore it for the case of educational data.

2.1 Basic Setting

In this work we focus on computing item similarities using learners' performance data. As Figure 1 shows, the similarity computation can also utilize information from domain experts or automatically determined information based on the inner structure of items (e.g., text of questions or some available meta-data).

We discuss different possibilities for computation of item similarities. Note that in our discussion we consistently use "similarity measures" (higher values correspond to higher similarity), some related works provide formulas for dissimilarity measures (distance of items; lower values correspond to higher similarity). This is just a technical issue, as we can easily transform similarity into dissimilarity by subtraction.

The input to item similarity computation are data about learner performance, i.e., a matrix $L \times I$, where L is the number of learners and I is the number of items. The matrix values specify learners' performance. The matrix is typically very sparse (many missing values). The output of the computation is an item similarity matrix, which specifies similarity for each pair of items.

Note that in our discussion we mostly ignore the issue of learning (change of learners skill as they progress through items). When learning is relatively slow and items are presented in a randomized order, learning is just a reasonably small source of noise and does not have a fundamental impact on the computation of item similarities. In cases where learning is fast or items are presented in a fixed order, it may be necessary to take learning explicitly into account.

2.2 Correctness of Answers

The basic type of information available in educational systems is the correctness of learners' answers. So we start with

similarity measures that utilize only this type of information, i.e., dichotomous data (correct/incorrect) on learners' answers on items. The advantage of these measures is that they are applicable in wide variety of settings.

With dichotomous data we can summarize learners' performance on items i and j using an agreement matrix with just four values (Table 1). Although we have just four values to quantify the similarity of items i and j, previous research has identified large number of different measures for dichotomous data and analyzed their relations [5, 12, 20]. For example Choi et al. [5] discuss 76 different measures, albeit many of them are only slight variations on one theme. Similarity measures over dichotomous data are often used in biology (co-occurrence of species) [14]. A more directly relevant application is the use of similarity measures for recommendations [30]. Recommender systems typically use either Pearson correlation or cosine similarity for computation of item similarities [11], but they consider richer than binary data.

Table 1: An agreement matrix for two items and definitions of similarity measures based on the agreement matrix ($n = a + b + c + d$ is the total number of observations).

		item i	
		incorrect	correct
item j	incorrect	a	b
	correct	c	d

Yule	$S_y = (ad - bc)/(ad + bc)$
Pearson	$S_p = (ad - bc)/\sqrt{(a+b)(a+c)(b+d)(c+d)}$
Cohen	$S_c = (P_o - P_e)/(1 - P_e)$ $P_o = (a + d)/n$ $P_e = ((a + b)(a + c) + (b + d)(c + d))/n^2$
Sokal	$S_s = (a + d)/(a + b + c + d)$
Jaccard	$S_j = a/(a + b + c)$
Ochiai	$S_o = a/\sqrt{(a + b)(a + c)}$

Table 1 provides definitions of 6 measures that we have chosen for our comparison. In accordance with previous research (e.g., [5, 14]) we call measures by names of researchers who proposed them. The choice of measures was done in such a way as to cover measures used in the most closely related work and measures which achieved good results (even if the previous work was in other domains). We also tried to cover different types of measures.

Pearson measure is the standard Pearson correlation coefficient evaluated over the dichotomous data. In the context of dichotomous data it is also called Phi coefficient or Matthews correlation coefficient. *Yule* measure is similar measure, which achieved good results in previous work [30]. *Cohen* measure is typically used as a measure of inter-rater agreement (it is more commonly called "Cohen's kappa"). In our setting it makes sense to consider this measure when we view learners' answers as "ratings" of items. Relations between these three measures are discussed in [32].

Ochiai coefficient is typically used in biology [14]. It is also equivalent to cosine similarity evaluated over dichotomous data; cosine similarity is often used in recommender systems for computing item similarity, albeit typically over interval data [7]. *Sokal* measure is also called Sokal-Michener or "simple matching". It is equivalent to accuracy measure used in information retrieval. Together with *Jaccard* measure they are often used in biology, but they have also been used for clustering of educational data [12].

Note that some similarity measures are asymmetric with respect to 0 and 1 values. These measures are typically used in contexts where the interpretation of binary values is presence/absence of a specific feature (or observation). In the educational context it is more natural to use measures which treat correct and incorrect answers symmetrically. Nevertheless, for completeness we have included also some of the commonly used asymmetric measures (Ochiai and Jaccard). In these cases we focus on incorrect answers (value a as opposed to d) as these are typically less frequent and thus bear more information.

2.3 Other Data Sources

The correctness of answers is the basic source of information about item similarities, but not the only one. We can also use other data. The second major type of performance data is response time (time taken to answer an item). The basic approach to utilization of response time is to combine it with the correctness of an answer. Given the correctness value $c \in \{0, 1\}$, a response time $t \in \mathbb{R}^+$, and the median of all response times τ, we combine them into a single score r. Examples of such transformations are: linear transformation for correct answers only ($r = c \cdot max(1 - t/2\tau, 0)$); exponential discounting used in MatMat [28] ($r = c \cdot min(1, 0.9^{t/\tau - 1})$); linear transformation inspired by *high speed, high stakes scoring rule* used in Math Garden [16] ($r = (2c - 1) \cdot max(1 - t/2\tau, 0)$). The first approach was used in our experiment due to its simplicity and high influence of response time information.

The scores obtained in this way are real numbers. Given the scores it is natural to compute similarity of two items using Pearson correlation coefficient of scores (over learners who answered both items). It is also possible to utilize specific wrong answers for computation of item similarity [25].

It is also possible to combine performance based measures with other types of data. For example we may estimate item similarity based on analysis of the content of items (syntactical similarity of texts), or collect expert opinion (manual categorization of items into several groups). The advantage of the similarity approach (compared to model based approach) is that different similarity measures can be usually combined in straightforward way by using a weighted average of different measures.

2.4 Second Level of Item Similarity

The basic computation of item similarities computes similarity of items i and j using only data about these two items. To improve a similarity measure, it is possible to employ a

"second of level of item similarity" that is based on the computed item similarity matrix and uses information on all items. Examples of such a second step is Euclidean distance or correlation. Similarity of items i and j is given by the Euclidean distance or Pearson correlation of rows i and j in the similarity matrix. Note that Euclidean distance may be used implicitly when we use standard implementation of some clustering algorithms (e.g., k-means).

With the basic approach to item similarity, we consider items similar when performance of learners on these items is similar. With the second step of item similarity, we consider two items similar when they behave similarly with respect to other items. The main reason for using this second step is the reduction of noise in data by using more information. This may be useful particularly to deal with learning. Two very similar items may have rather low direct similarity, because getting a feedback on the first item can strongly influence the performance on the second item. However, we expect both items to have similar similarities to other items.

A more technical reason to using the second step (particularly the Euclidean distance) is to obtain a measure that is a distance metric. The measures described above mostly do not satisfy triangle inequality and thus do not satisfy the requirements on distance metric; this property may be important for some clustering algorithms.

3. EVALUATION

In this work we focus on item similarity, but we keep the overall context depicted in Figure 1 in mind. The quality of a visualization is to a certain degree subjective and difficult to quantify, but the quality of clusters can be quantified and thus we can use it to compare similarity measures. From the large pool of existing clustering algorithms [15] we consider k-means, which is the most common implementation of centroid-based clustering, and hierarchical clustering. We used agglomerative or "bottom up" approach where items are successively merged to clusters using Ward's method as linkage criteria.

3.1 Data

We use data from real educational systems as well as simulated learner data. Real-world data provide information about the realistic performance of techniques, but the evaluation is complicated by the fact that we do not know the "ground truth" (the "correct" similarity or clusters of items). Simulated data provide a setting that is in many aspects simplified but allows easier evaluation thanks to the access to the ground truth.

For generating simulated data we use a simple approach with minimal number of assumptions and ad hoc parameters. Each item belongs to one of k knowledge components. Each knowledge component contains n items. Each item has a difficulty generated from the standard normal distribution $d_i \sim \mathcal{N}(0, 1)$. Skills of learners with respect to individual knowledge components are independent. Skill of a learner l with respect to knowledge component j is generated from the standard normal distribution $\theta_{lj} \sim \mathcal{N}(0, 1)$. We assume no learning (constant skills). Answers are generated as Bernoulli trials with the probability of a correct answer given by the logistic function of the difference of a relevant skill and an item difficulty (a Rasch model): $p = exp(\theta_{lj} - d_i)^{-1}$. This approach is rather standard, for example Piech at al. [26] use very similar procedure and also other works use closely related procedures [4, 12]. In the experiment reported below the basic setting is 100 learners, 5 knowledge components with 20 items each.

Table 2: Data used for analysis.

	learners	items	answers
Czech 1 (adjectives)	1 134	108	62 613
Czech 2	4 567	210	336 382
MatMat: numbers	6 434	60	67 753
MatMat: addition	3 580	135	20 337
Math Garden: addition	83 297	30	881 994
Math Garden: multiplic.	97 842	30	1 233 024

To evaluate techniques on realistic educational data, we use data from three educational systems. Table 2 describes the size of the used data sets.

Umíme Česky (`umimecesky.cz`) is a system for practice of Czech spelling and grammar. We use data only from one exercise from the system – simple "fill-in-the-blank" questions with two options. We use only data on the correctness of answers (response time is available, but since it depends on the text of a particular item its utilization is difficult). We focus particularly on one subset of items: questions about the choice between i/y in suffixes of Czech adjectives. For this subset we have manually determined 7 groups of items corresponding to Czech grammar rules.

MatMat (`matmat.cz`) is a system for practice of basic arithmetic (e.g., counting, addition, multiplication). For each item we know the underlying construct (e.g., "13" or "7 + 8") and also the specific form of questions (e.g., what type of visualization has been used). We use data on both correctness and response time. We selected the two largest subsets: multiplication and numbers (practice of number sens, counting).

Math Garden is another system for practice of basic arithmetic [16]. This system is more widely used than MatMat, but we do not have direct access to the system and detailed data. For the analysis we reuse publicly available data from previous research [6]. The available data contain both correctness of answers and response times, but they contain information only about 30 items without any identification of these items.

3.2 Comparison of Similarity Measures

To evaluate similarity measures we consider several types of analysis. With simulated data, we analyze the similarity measures with respect to the ground truth while for real-world data we evaluate correlations among similarity measures. We also compare the quality of subsequent clusterings using adjusted Rand index (ARI) [27, 31], which measures the agreement of two clusterings (with a correction for agreement due to chance). Typically, we use the adjusted Rand index to compare the clustering with a ground truth (available for simulated data) or with a manually provided

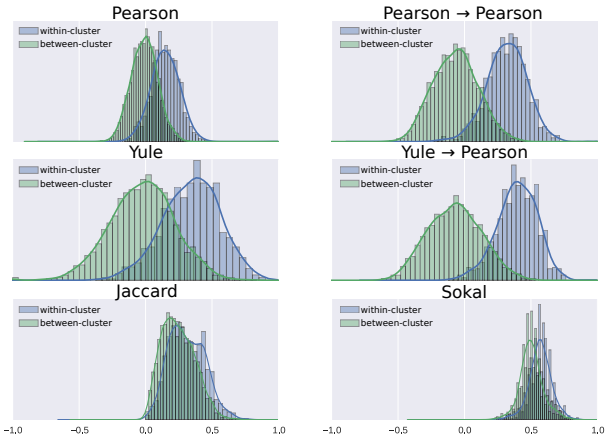


Figure 2: Differences between similarity values inside knowledge components and between them. Simulated data set with the basic setting were used.

classification (available for the Czech 1 data set). It can be also used to compare two detected clusterings (clusterings based on two different algorithms or clusterings based on two independent halves of data).

As a first step in the evaluation of similarity measures, we consider experiments with simulated data where we can utilize the ground truth. In clustering we expect high within-cluster similarity values and low between-cluster similarity values. Figure 2 shows distribution of the similarity values for selected measures and suggest which measures separate within-cluster and between-cluster values better and therefore which measures will be more useful in clustering. The results show that for Jaccard and Sokal measures the values overlap to a large degree, whereas Pearson and Yule measures provide better results. Adding the second step – Pearson correlation in this example – to the similarity measure separates within-cluster and between-cluster values better. That suggests that extending similarities in this way is not only necessary step for some subsequent algorithms such as k-means but also a useful technique with better performance.

For data coming from real systems we do not know the ground truth and thus we can only compare the similarity measures to each other. To evaluate how similar two measures are we take all similarity values for all item pairs and computed correlation coefficient. Figure 3 shows results for two data sets which are good representatives of overall results. Pearson and Cohen measures are highly correlated (> 0.98) across all data sets and have nearly the same values (although not exactly the same). Larger differences (but only up to 0.1) can be found typically when one of the values in the agreement matrix is small and that happens only for poorly correlated items with the resulting similarity value around 0. The second pair of highly correlated measures is Ochiai and Jaccard, which are both asymmetric with respect to the agreement matrix. The correlation between these two pairs of measures vary depending on data set and in some cases drops up to 0.5. Because of this high correlation within these pairs we further report results only

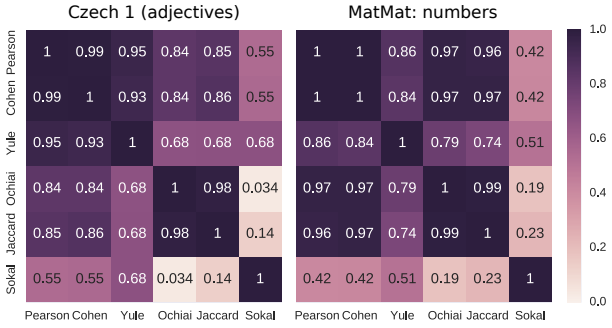


Figure 3: Correlations of similarity measures.

for Pearson and Jaccard measures. Yule measure is usually similar to Pearson measure (correlation usually around 0.9). The main difference is that the Yule measure spreads values more evenly across the interval [-1, 1]. Sokal is the most outlying measure with no correlation or small correlation (usually < 0.6) with all other measures.

Figure 4 shows the effect of the second levels of item similarity on the Pearson measure (results for other measures are analogical). The Euclid distance as second level similarity brings larger differences (lower correlation) than Pearson correlation. The correlations for large data sets such as Math Garden are usually high (> 0.9) and conversely the lowest correlations are found in results for small data sets. This suggests that the second level of similarity is more significant, and thus potentially more useful, where only limited amount of data is available.

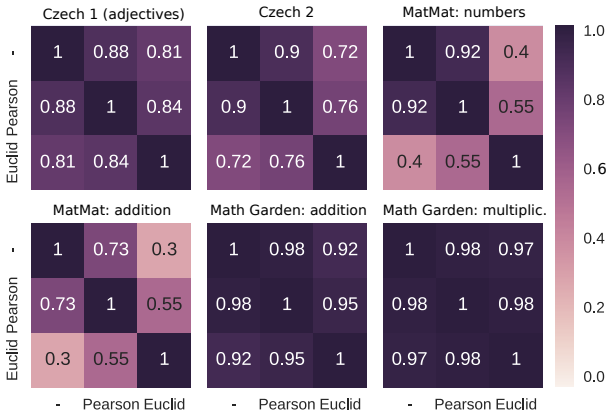


Figure 4: Correlations of Pearson measure and Pearson with different second levels.

Finally, we evaluate the quality of the similarity measures according to the performance of the subsequent clustering. From the two considered clustering methods we used the hierarchical clustering in this comparison because it naturally works with similarity measure and does not require metric space. The other two methods have similar result with same conclusions. Table 3 and Figure 5 show results. Although the results are dependent on the specific data set and the used clustering algorithm, there is quite clear general conclusion. Pearson and Yule measures provide better results

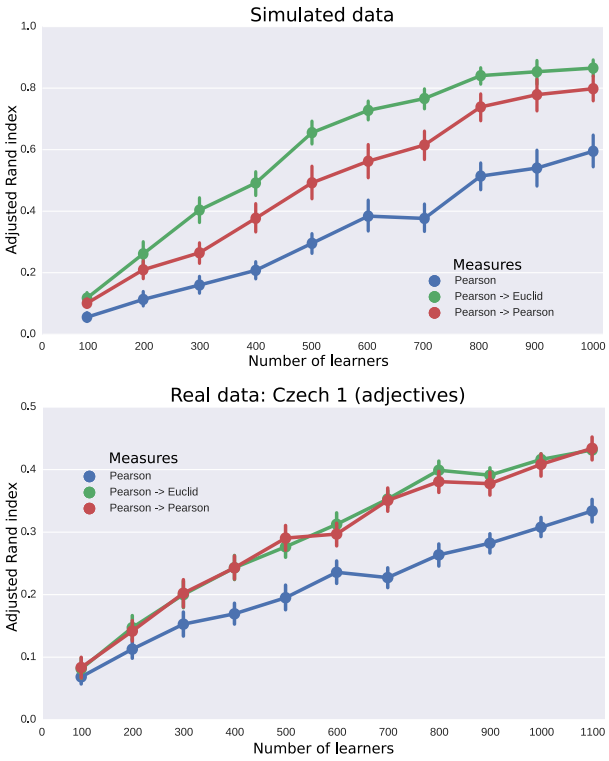


Figure 5: The quality of clustering for different measures used in the second step of item similarity. Top: Simulated data with 5 correlated skills. Bottom: Czech grammar with 7 manually determined clusters.

than Jaccard and Sokal, i.e., for the considered task the later two measures are not suitable. The Pearson is usually slightly better than Yule but the choice between them seems not to be fundamental (which is not surprising given that they are highly correlated). The results also show that the "second step" is always useful. The result for simulated data favor Euclidean distance over Pearson but there are almost no differences for real-world data.

3.3 Do We Have Enough Data?

In machine learning the amount of available data often is more important than the choice of a specific algorithm [2]. Our results suggest that once we choose a suitable type of similarity measure (e.g., Pearson, Cohen, or Yule), the differences between these measures are not fundamental, the more important issue becomes the size of available data.

Specifically, for a given data set we want to know whether the data are sufficiently large so that the computed item similarities are meaningful and stable. This issue can be explored by analyzing confidence intervals for computed similarity values. As a simple approach to analysis of similarity stability we propose the following approach: We split the available data into two independent halves (in a learner stratified manner), for each half we compute the item similarities, and we compute the correlation of the resulting item similarities.

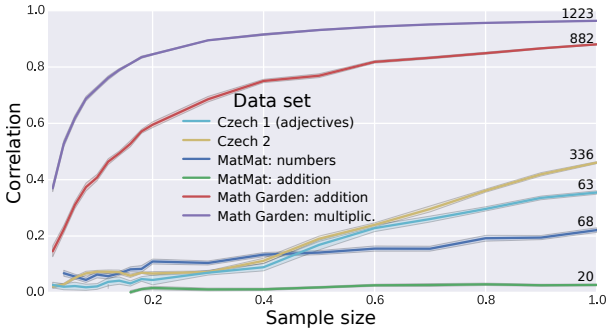


Figure 6: Stability of similarity measure (Yule) for real-world data sets. Data set was sampled, split to halves and Pearson correlation was computed for similarity values. Numbers on the right side indicate thousands of answers in data sets.

We can also perform this computation for artificially reduced data sets – this shows how the stability of results increases with the size of data. Figure 6 shows this kind of analysis for our data (real-world data sets). We clearly see large differences among individual data sets. Math Garden data set contains large number of answers and only a few items, the results show excellent stability, clearly in this case we have enough data to analyze item similarities. For the Czech grammar data set we have large number of answers, but these are divided among relatively large number of items. The results show a reasonably good stability, the data are usable for analysis, but clearly more data can bring improvement. For MatMat data the stability is poor, to draw solid conclusions about item similarities we need more data.

3.4 Response Time Utilization

The incorporation of response time information to similarity measure can change the meaning of similarity. Figure 7 gives such example and shows projection of items from MatMat practicing number sense. Similar items according to measures using only correctness of answers tend to be items with the same graphical representation in the system. On the other hand, similar items according to measures using also response time are usually items practicing close numbers.

We used this method also on data sets from Math Garden, which are much larger. In this case the use of response times has only small impact on the computed item similarities (correlations between 0.9 and 0.95). However, the use of response times influences how quickly does the computation converge, i.e., how much data do we need. To explore this we consider as the ground truth the average of computed similarity matrices with and without response times for the whole data set. Then we used smaller samples of the data set, used them to compute item similarities and checked the agreement with this ground truth. Figure 8 shows the difference between speed of convergence of measure with and without response time utilization. Results shows that the measure which use addition information from response time converges to ground truth much faster. This result suggests that the use of response time can improve clustering or visualizations when only small number of answers are available.

Table 3: Comparison of similarity measures for one real-world data (with sampled students) set and simulated data sets with c knowledge components and l learners. The values provide the adjusted Rand index (with 0.95 confidence interval) for a hierarchical clustering computed based on the specific similarity measure. The top result for every data set is highlighted.

	Czech 1 (c=7)	$l = 50, c = 5$	$l = 100, c = 5$	$l = 200, c = 5$	$l = 100, c = 2$	$l = 100, c = 10$
Pearson	0.32 ± 0.02	0.26 ± 0.04	0.48 ± 0.05	0.84 ± 0.05	0.77 ± 0.12	0.34 ± 0.04
Jaccard	0.31 ± 0.03	0.06 ± 0.03	0.15 ± 0.04	0.29 ± 0.08	0.32 ± 0.18	0.09 ± 0.02
Yule	0.31 ± 0.03	0.19 ± 0.04	0.43 ± 0.05	0.77 ± 0.07	0.60 ± 0.15	0.31 ± 0.03
Sokal	0.15 ± 0.06	0.11 ± 0.02	0.18 ± 0.03	0.25 ± 0.05	0.12 ± 0.11	0.14 ± 0.02
Pearson → Euclid	$\mathbf{0.43} \pm 0.01$	$\mathbf{0.45} \pm 0.05$	$\mathbf{0.80} \pm 0.06$	$\mathbf{0.98} \pm 0.01$	$\mathbf{0.95} \pm 0.03$	$\mathbf{0.67} \pm 0.04$
Yule → Euclid	0.32 ± 0.02	0.36 ± 0.05	0.65 ± 0.07	0.94 ± 0.04	0.89 ± 0.11	0.43 ± 0.03
Pearson → Pearson	0.41 ± 0.03	0.39 ± 0.05	0.73 ± 0.06	0.96 ± 0.02	0.92 ± 0.03	0.55 ± 0.04
Yule → Pearson	0.32 ± 0.03	0.38 ± 0.05	0.72 ± 0.06	0.97 ± 0.02	0.94 ± 0.04	0.55 ± 0.05

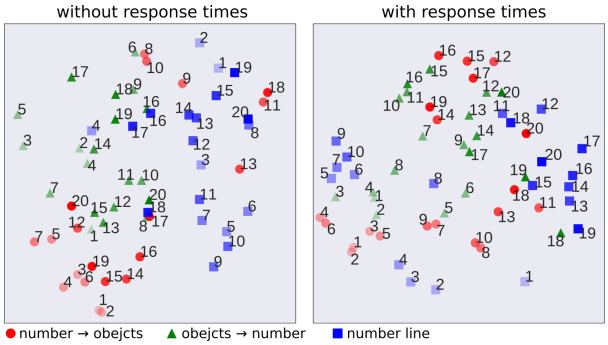


Figure 7: Projection of items practicing number sense from MatMat system. Left: Measure based only correctness. Right: Measure using response time. Opacity corresponds to the number value of the item and color corresponds to the graphical representation of the task.

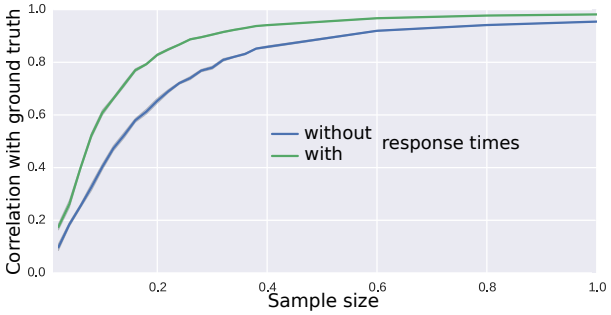


Figure 8: The speed of convergence to ground truth for measures with and without response time on Math Garden addition data set.

4. DISCUSSION

Our focus is the automatic computation of item similarities based on learners' performance data. These similarities can be then used in further analysis of an item relations such as an item clustering or a visualization. This outlines direction for future work in which methods using the item similarities should be studied in more detail. Compared to alternative approaches that have been proposed for the task (e.g., matrix factorizations, neural networks), the item similarity approach is rather straightforward, easy to realize, and it can be easily combined with other sources of information about items (text of items, expert opinion). For these reasons the item similarity approach should be used at least as a baseline in proposals for more complex methods like deep knowledge tracing [26].

The most difficult step in this approach is the choice of a similarity measure. Once we make a specific choice, the realization of the approach is easy. Our results provide some guidelines for this choice. Pearson, Yule, and Cohen measures lead to significantly better results than Ochiai, Sokal, and Jaccard measures. It is also beneficial to use the second step of item similarity (e.g., the Euclidean distance over vectors of item similarities). The exact choice of details does not seem to make fundamental difference (e.g., Pearson versus Yule in the first step, the Euclidean distance versus Pearson correlation in the second step). The Pearson correlation coefficient is a good "default choice", since it provides quite robust results and is applicable in several settings and steps. It also has the pragmatic advantage of having fast, readily available implementation in nearly all computational environments, whereas measures like Yule may require additional implementation effort.

The amount of data available is the critical factor for the success of automatic analysis of item relations. A key question for practical applications is thus: "Do we have enough data to use automated techniques?" In this work we used several specific methods for analysis of this question, but the issue requires more attention – not just for the item similarity approach, but also for other methods proposed in previous work. For example previous work on deep knowledge tracing [26], which studies closely related issues, states only that deep neural networks require large data without providing any specific quantification what 'large' means. The necessary quantity of data is, of course, connected to the quality of data – some data sources are more noisy than other, e.g., answers from voluntary practice contain more noise than answers from high-stakes testing. An important direction for future work is thus to compare model based and item simi-

larity approaches while taking into account the 'amount and quality of data available' issue.

5. REFERENCES

[1] R. S. Baker. Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, 26(2):600–614, 2016.

[2] M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *Proc. of Association for Computational Linguistics*, pages 26–33, 2001.

[3] T. Barnes. The q-matrix method: Mining student response data for knowledge. In *Educational Data Mining Workshop*, 2005.

[4] W.-H. Chen and D. Thissen. Local dependence indexes for item pairs using item response theory. *Journal of Educational and Behavioral Statistics*, 22(3):265–289, 1997.

[5] S.-S. Choi, S.-H. Cha, and C. C. Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.

[6] F. Coomans, A. Hofman, M. Brinkhuis, H. L. van der Maas, and G. Maris. Distinguishing fast and slow processes in accuracy-response time data. *PloS one*, 11(5):e0155149, 2016.

[7] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[8] M. C. Desmarais. Mapping question items to skills with non-negative matrix factorization. *ACM SIGKDD Explorations Newsletter*, 13(2):30–36, 2012.

[9] M. C. Desmarais and R. S. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[10] M. C. Desmarais, B. Beheshti, and P. Xu. The refinement of a q-matrix: Assessing methods to validate tasks to skills mapping. In *Proc. of Educational Data Mining*, pages 308–311, 2014.

[11] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer, 2011.

[12] H. Finch. Comparison of distance measures in cluster analysis with dichotomous data. *Journal of Data Science*, 3(1):85–100, 2005.

[13] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[14] D. A. Jackson, K. M. Somers, and H. H. Harvey. Similarity coefficients: measures of co-occurrence and association or simply measures of occurrence? *American Naturalist*, pages 436–453, 1989.

[15] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[16] S. Klinkenberg, M. Straatemeier, and H. Van der Maas. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824, 2011.

[17] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science*, 36(5):757–798, 2012.

[18] Y. Koren and R. Bell. Advances in collaborative filtering. *Recommender Systems Handbook*, pages 145–186, 2011.

[19] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *Journal of Machine Learning Research*, 15(1):1959–2008, 2014.

[20] S.-F. M. Liang and L.-W. Tzeng. Assessing suitability of similarity coefficients in measuring human mental models. In *Network of Ergonomics Societies Conference*, pages 1–5. IEEE, 2012.

[21] J. Nižnan, R. Pelánek, and J. Řihák. Using problem solving times and expert opinion to detect skills. In *Proc. of Educational Data Mining*, pages 434–434, 2014.

[22] J. Nižnan, R. Pelánek, and J. Řihák. Student models for prior knowledge estimation. In *Proc. of Educational Data Mining*, pages 109–116, 2015.

[23] M. O'Connor and J. Herlocker. Clustering items for collaborative filtering. In *Proc. of the ACM SIGIR Workshop on Recommender Systems*, volume 128. UC Berkeley, 1999.

[24] Y.-J. Park and A. Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proc. of Recommender systems*, pages 11–18. ACM, 2008.

[25] R. Pelánek and J. Řihák. Properties and applications of wrong answers in online educational systems. In *Proc. of Educational Data Mining*, 2016.

[26] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015.

[27] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[28] J. Rihák. Use of time information in models behind adaptive system for building fluency in mathematics. In *Proc. of Educational Data Mining*, 2015.

[29] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proc. of Computer and Information Technology*, volume 1, 2002.

[30] E. Şenyürek and H. Polat. Effects of binary similarity measures on top-n recommendations. *Anadolu University Journal of Science and Technology – A Applied Sciences and Engineering*, 14(1):55–65, 2013.

[31] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proc. of Machine Learning*, pages 1073–1080. ACM, 2009.

[32] M. J. Warrens. On association coefficients for 2×2 tables and properties that do not depend on the marginal distributions. *Psychometrika*, 73(4):777–789, 2008.

Adaptive Sequential Recommendation for Discussion Forums on MOOCs using Context Trees

Fei Mi Boi Faltings
Artificial Intelligence Lab
École polytechnique fédérale de Lausanne, Switzerland
firstname.lastname@epfl.ch

ABSTRACT

Massive open online courses (MOOCs) have demonstrated growing popularity and rapid development in recent years. Discussion forums have become crucial components for students and instructors to widely exchange ideas and propagate knowledge. It is important to recommend helpful information from forums to students for the benefit of the learning process. However, students or instructors update discussion forums very often, and the student preferences over forum contents shift rapidly as a MOOC progresses. So, MOOC forum recommendations need to be adaptive to these evolving forum contents and drifting student interests. These frequent changes pose a challenge to most standard recommendation methods as they have difficulty adapting to new and drifting observations. We formalize the discussion forum recommendation problem as a sequence prediction problem. Then we compare different methods, including a new method called context tree (CT), which can be effectively applied to online sequential recommendation tasks. The results show that the CT recommender performs better than other methods for MOOCs forum recommendation task. We analyze the reasons for this and demonstrate that it is because of better adaptation to changes in the domain. This highlights the importance of considering the adaptation aspect when building recommender system with drifting preferences, as well as using machine learning in general.

Keywords

MOOCs forum recommendation, context tree, model adaptation

1. INTRODUCTION

With the increased availability of data, machine learning has become the method of choice for knowledge acquisition in intelligent systems and various applications. However, data and the knowledge derived from it have a timeliness, such that in a dynamic environment not all the knowledge acquired in the past remains valid. Therefore, machine learning models should acquire new knowledge incrementally and adapt to the dynamic environments. Today, many intelligent systems deal with dynamic environments: information on websites, social networks, and applications in commercial markets. In such evolving environments, knowledge needs to adapt to the changes very frequently. Many statistical machine learning techniques interpolate between input data and thus their models can adapt only slowly to new situations. In this paper, we consider the dynamic environments for recommendation task. Drifting user interests and preferences [3, 11] are important in building personal assistance systems, such as recommendation systems for social networks or for news websites where recommendations need be adaptive to drifting trends rather than recommending obsolete or well-known information. We focus on the application of recommending forum contents for massive open online courses (MOOCs) where we found that the adaptation issue is a crucial aspect for providing useful and trendy information to students.

The rapid emergence of some MOOC platforms and many MOOCs provided on them has opened up a new era of education by pushing the boundaries of education to the general public. In this special online classroom setting, sharing with your classmates or asking help from instructors is not as easy as in traditional brick-and-mortar classrooms. So discussion forums there have become one of the most important components for students to widely exchange ideas and to obtain instructors' supplementary information. MOOC forums play the role of social learning media for knowledge propagation with increasing number of students and interactions as a course progresses. Every member in the forum can talk about course content with each other, and the intensive interaction between them supports the knowledge propagation between members of the learning community.

The online discussion forums are usually well structured via the different threads which are created by students or instructors; they can contain several posts and comments within the topic. An example of the discussion forum from a famous "Machine Learning" course by Andre Ng on Coursera[1] is shown in Figure 1. The left figure shows various threads and the right figure illustrates some replies within the last thread ("Having a problem with the Collaborative Filtering Cost"). In general, the replies within a thread are related to the topic of the thread and they can also refer to some other threads for supplementary information, like the link in the second reply. Our goal is to point the students towards useful forum threads through effectively mining forum visit patterns.

Two aspects set forum recommendation system for MOOCs apart from other recommendation scenarios. First, student interests and preferences drift fast during the span of a course, which is influenced by the dynamics in forums and the content of the course; second, the pool of items to be recommended and the items them-

[1] https://www.coursera.org/

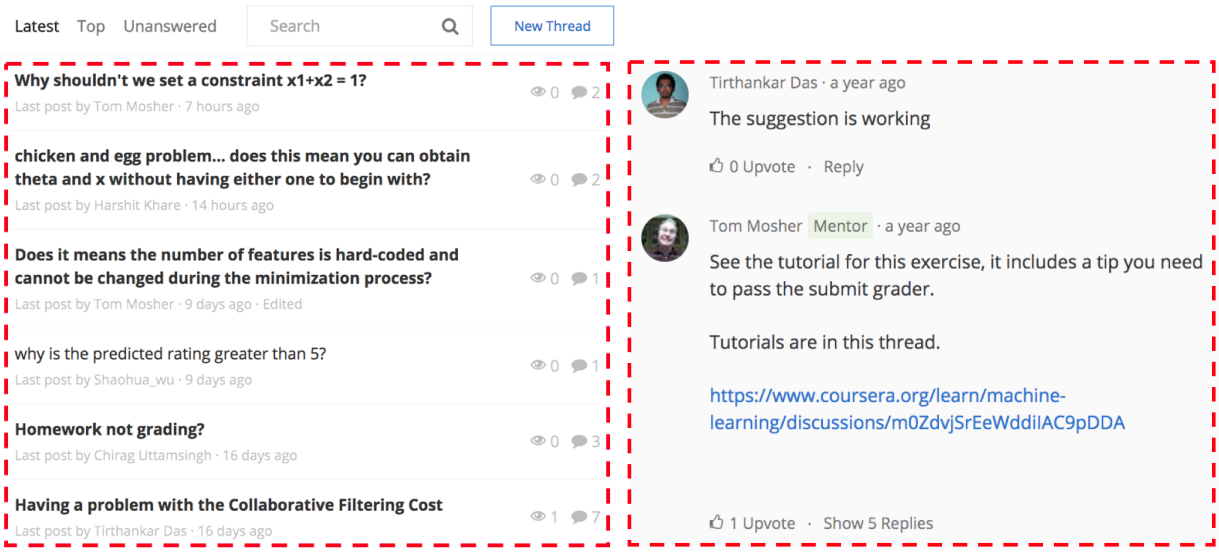


Figure 1: An sample discussion forum. Left: sample threads. Right: replies within the last thread ("Having a problem with the Collaborative Filtering Cost").

selves are evolving over time because forum threads can be edited very frequently by either students or instructors. So the recommendations provided to students need to be adaptive to these drifting preferences and evolving items. Traditional recommendation techniques, such as collaborative filtering and methods based on matrix factorization, only adapt slowly, as they build an increasingly complex model of users and items. Therefore, when a new item is superseded by a newer version or a new preference pattern appears, it takes time for recommendations to adapt. To better address the dynamic nature of recommendation in MOOCs, we model the recommendation problem as a dynamic and sequential machine learning problem for the task of predicting the next item in a sequence of items consumed by a user. During the sequential process, the challenge is combining old knowledge with new knowledge such that both old and new patterns can be identified fast and accurately. We use algorithms for sequential recommendation based on variable-order Markov models. More specifically, we use a structure called context tree (CT) [21] which was originally proposed for lossless data compression. We apply the CT method for recommending discussion forum contents for MOOCs, where adapting to drifting preferences and dynamic items is crucial. In experiments, it is compared with various sequential and non-sequential methods. We show that both old knowledge and new patterns can be captured effectively through context activation using CT, and that this is why it is particularly strong at adapting to drifting user preferences and performs extremely well for MOOC forum recommendation tasks.

The main contribution of this paper is fourfold:

- We applied the context tree structure to a sequential recommendation tasks where dynamic item sets and drifting user preferences are of great concern.
- Analyze how the dynamic changes in user preferences are followed in different recommendation techniques.
- Extensive experiments are conducted for both sequential and non-sequential recommendation settings. Through the experimental analysis, we validate our hypothesis that the CT recommender adapts well to drifting preferences.
- Partial context matching (PCT) technique, built on top of the standard CT method, is proposed and tested to generalize to new sequence patterns, and it further boosts the recommendation performance.

2. RELATED WORK

Typical recommender systems adopt a static view of the recommendation process and treat it as a prediction problem over all historical preference data. From the perspective of generating adaptive recommendations,we contend that it is more appropriate to view the recommendation problem as a sequential decision problem. Next, we mainly review some techniques developed for recommender systems with temporal or sequential considerations.

The most well-known class of recommender system is based on collaborative filtering (CF) [19]. Several attempts have been made to incorporate temporal components into the collaborative filtering setting to model users' drifting preferences over time. A common way to deal with the temporal nature is to give higher weights to events that happened recently. [6, 7, 15] introduced algorithms for item-based CF that compute the time weightings for different items by adding a tailored decay factor according to the user's own purchase behavior. For low dimensional linear factor models, [11] proposed a model called "TimeSVD" to predict movie ratings for Netflix by modeling temporal dynamics, including periodic effects, via matrix factorization. As retraining latent factor models is costly, one alternative is to learn the parameters and update the decision function online for each new observation [1, 16]. [10] applied the online CF method, coupled with an item popularity-aware weighting scheme on missing data, to recommending social web contents with implicit feedbacks.

Markov models are also applied to recommender systems to learn the transition function over items. [24] treated recommendation as a univariate time series problem and described a sequential model with a fixed history. Predictions are made by learning a forest of decision trees, one for each item. When the number of items is big, this approach does not scale. [17] viewed the problem of generating recommendations as a sequential decision problem and they con-

sidered a finite mixture of Markov models with fixed weights. [4] applied Markov models to recommendation tasks using skipping and weighting techniques for modeling long-distance relationships within a sequence. A major drawback of these Markov models is that it is not clear how to choose the order of Markov chain.

Online algorithms for recommendation are also proposed in several literatures. In [18], a Q-learning-based travel recommender is proposed, where trips are ranked using a linear function of several attributes and the weights are updated according to user feedback. A multi-armed bandit model called LinUCB is proposed by [13] for news recommendation to learn the weights of the linear reward function, in which news articles are represented as feature vectors; click-through rates of articles are treated as the payoffs. [20] proposed a similar recommender for music recommendation with rating feedback, called Bayes-UCB, that optimizes the nonlinear reward function using Bayesian inference. [14] used a Markov Decision Process (MDP) to model the sequential user preferences for recommending music playlists. However, the exploration phase of these methods makes them adapt slowly. As user preferences drift fast in many recommendation setting, it is not effective to explore all options before generating useful ones.

Within the context of recommendation for MOOCs, [23] proposed an adaptive feature-based matrix factorization framework for course forum recommendation, and the adaptation is achieved by utilizing only recent features. [22] designed a context-aware matrix factorization model to predict student preferences for forum contents, and the context considered includes only supplementary statistical features about students and forum contents. In this paper, we focus on a class of recommender systems based on a structure, called context tree [21], which was originally used to estimate variable-order Markov models (VMMs) for lossless data compression. Then, [2, 12, 5] applied this structure to various discrete sequence prediction tasks. Recently it was applied to news recommendation by [8, 9]. The most important property of online algorithms is the no-regret property, meaning that the model learned online is eventually as good as the best model that could be learned offline. According to [21], the no-regret property is achieved by context trees for the data compression problem. Regret analysis for CT was conducted through simulation by [5] for stochastically generated hidden Markov models with small state space. They show that CT achieves the no-regret property when the environment is stationary. As we focus on dynamic recommendation environments with time-varying preferences and limited observations, the no-regret property can be hardly achieved while the model adaptation is a bigger issue for better performance.

3. CONTEXT TREE RECOMMENDER

Due to the sequential item consumption process, user preferences can be summarized by the last several items visited. When modeling the process as a fixed-order Markov process [17], it is difficult to select the order. A variable-order Markov model (VMM), like a context tree, alleviates this problem by using a context-dependent order. The context tree is a space efficient structure to keep track of the history in a variable-order Markov chain so that the data structure is built incrementally for sequences that actually occur. A local prediction model, called expert, is assigned to each tree node, it only gives predictions for users who have consumed the sequence of items corresponding to the node. In this section, we first introduce how to use the CT structure and the local prediction model for sequential recommendation. Then, we discuss adaptation properties and the model complexity of the CT recommender.

3.1 The Context Tree Data Structure

In CT, a sequence $\mathbf{s} = \langle n_1, \ldots, n_l \rangle$ is an ordered list of items $n_i \in N$ consumed by a user. The sequence of items viewed until time t is $\mathbf{s}_t$ and the set of all possible sequences $\mathcal{S}$.

A context $S = \{\mathbf{s} \in \mathcal{S} : \xi \prec \mathbf{s}\}$ is the set of all possible sequences in $\mathcal{S}$ ending with the suffix ξ. ξ is the suffix ($\prec$) of $\mathbf{s}$ if last elements of $\mathbf{s}$ are equal to ξ. For example, one suffix ξ of the sequence $\mathbf{s} = \langle n_2, n_3, n_1 \rangle$ is given by $\xi = \langle n_3, n_1 \rangle$.

A context tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V}$ and edges $\mathcal{E}$ is a partition tree over all contexts of $\mathcal{S}$. Each node $i \in \mathcal{V}$ in the context tree corresponds to a context S_i. If node i is the ancestor of node j then $S_j \subset S_i$. Initially the context tree $\mathcal{T}$ only contains a root node with the most general context. Every time a new item is consumed, the active leaf node is split into a number of subsets, which then become nodes in the tree. This construction results in a variable-order Markov model. Figure 2 illustrates a simple CT with some sequences over an item set $\langle n_1, n_2, n_3 \rangle$. Each node in the CT corresponds to a context. For instance, the node $\langle n_1 \rangle$ represents the context with all sequences end with item n_1.

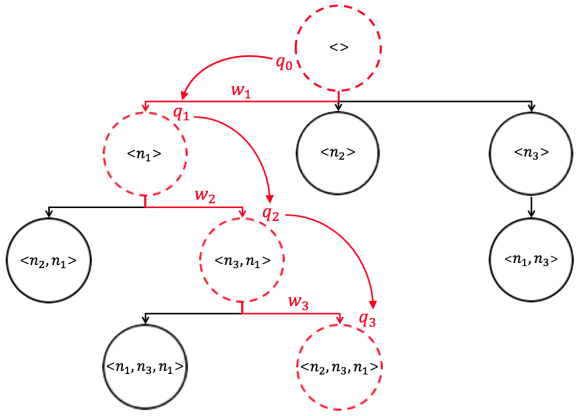


Figure 2: An example context tree. For the sequence $\mathbf{s} = \langle n_2, n_3, n_1 \rangle$, nodes in red-dashed are activated.

3.2 Context Tree for Recommendation

For each context S_i, an expert μ_i is associated in order to compute the estimated probability $\mathbb{P}(n_{t+1}|\mathbf{s}_t)$ of the next item n_{t+1} under this context. A user's browsing history $\mathbf{s}_t$ is matched to the CT and identifies a path of matching nodes (see Figure 2). All the experts associated with these nodes are called *active*. The set of *active* experts $\mathcal{A}(\mathbf{s}_t) = \{\mu_i : \xi_i \prec \boldsymbol{s}_t\}$ is the set of experts μ_i associated to contexts $S_i = \{\mathbf{s} : \xi_i \prec \mathbf{s}_t\}$ such that ξ_i are suffix of $\mathbf{s}_t$. $\mathcal{A}(\mathbf{s}_t)$ is responsible for the prediction for $\mathbf{s}_t$.

3.2.1 *Expert Model*

The standard way for estimating the probability $\mathbb{P}(n_{t+1}|\mathbf{s}_t)$, as proposed by [5], is to use a Dirichlet-multinomial prior for each expert μ_i. The probability of viewing an item x depends on the number of times α_{xt} the item x has been consumed when the expert is active until time t. The corresponding marginal probability is:

$$\mathbb{P}_i(n_{t+1} = x|\mathbf{s}_t) = \frac{\alpha_{xt} + \alpha_0}{\sum_{j \in \mathcal{N}} \alpha_{jt} + \alpha_0} \quad (1)$$

where α_0 is the initial count of the Dirichlet prior

3.2.2 Combining Experts to Prediction

When making recommendation for a sequence $\mathbf{s}_t$, we first identify the set of contexts and active experts that match the sequence. The predictions given by all the active experts are combined by mixing the recommendations given by them:

$$\mathbb{P}(n_{t+1} = x|\mathbf{s}_t) = \sum_{i \in \mathcal{A}(\mathbf{s}_t)} u_i(\mathbf{s}_t)\mathbb{P}_i(n_{t+1} = x|\mathbf{s}_t) \quad (2)$$

The mixture coefficient $u_i(\mathbf{s}_t)$ of expert μ_i is computed in Eq. 3 using the weight $w_i \in [0, 1]$. Weight w_i is the probability that the chosen recommendation stops at node i given that the it can be generated by the first i experts, and it can be updated in using Eq.5.

$$u_i(\mathbf{s}_t) = \begin{cases} w_i \prod_{j:S_j \subset S_i}(1 - w_j), & \text{if } \mathbf{s}_t \in S_i \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The combined prediction of the first i experts is defined as q_i and it can be computed using the recursion in Eq. 4. The recursive construction that estimates, for each context at a certain depth i, whether it makes better prediction than the combined prediction q_{i-1} from depth $i - 1$.

$$q_i = w_i\mathbb{P}_i(n_{t+1} = x|\mathbf{s}_t) + (1 - w_i)q_{i-1} \quad (4)$$

The weights are updated by taking into account the success of a recommendation. When a user consumes a new item x, we update the weights of the active experts corresponding to the suffix ending before x according to the probability $q_i(x)$ of predicting x sequentially via Bayes' theorem. The weights are updated in closed form in Eq. 5, and a detailed derivation can be found in [5].

$$w_i' = \frac{w_i\mathbb{P}_i(n_{t+1} = x|\mathbf{s}_t)}{q_i(x)} \quad (5)$$

3.2.3 CT Recommender Algorithm

The whole recommendation process first goes through all users' activity sequences over time incrementally to build the CT; the local experts and weights updated using Equations 1 and 5 respectively. As users browse more contents, more contexts and paths are added and updated, thus building a deeper, more complete CT. The recommendation for an activity or context in a sequence is generated using Eq. 2 continuously as experts and weights are updated. At the same time, a pool of candidate items is maintained through a dynamically evolving context tree. As new items are added, new branches are created. At the same time, nodes corresponding to old items are removed as soon as they disappear from the current pool.

The CT recommender is a mixture model. On the one hand, the prediction $\mathbb{P}(n_{t+1} = x|\mathbf{s}_t)$ is a mixture of the predictions given by all the activated experts along the activated path so that it's a mixtures of local experts or a mixture of variable order Markov models whose oder are defined by context depths. On the other hand, one path in a CT can be constructed or updated by multiple users so that it's a mixture of users' preferences.

3.3 Adaptation Analysis

Our hypothesis, which is validated in later experiments, is that the CT recommender can be applied elegantly to domains where adaptation and timeliness are of concern. Two properties of the CT methods are crucial to the goal. First, the model parameter learning process and recommendations generated are online such that the model adapts continuously to a dynamic environment. Second, adaptability can be achieved by the CT structure itself as knowledge is organized and activated by context. New items or paths are recognized in new contexts, whereas old items can still be accessed in their old contexts. It allows the model to make predictions using more complex contexts as more data is acquired so that old and new knowledge can be elegantly combined. For new knowledge or patterns added to an established CT, they can immediately be identified through context matching. This context organization and context matching mechanism help new patterns to be recognized to adapt to changing environments.

3.4 Complexity Analysis

Learning CT uses the recursive update defined in Eq. 4 and recommendations are generated by weighting the experts' predictions along the activated path given by Eq. 2. For trees of depth D, the time complexity of model learning and prediction for a new observation are both $O(D)$. For input sequence of length T, the updating and recommending complexity are $O(M^2)$, where $M = min(D, T)$. Space complexity in the worst case is exponential to the depth of the tree. However, as we do not generate branches unless the sequence occurs in the input, we achieve a much lower bound determined by the total size of the input. So the space complexity is $O(N)$, where N is the total number of observations. Compared with the way that Markov models are learned, in which the whole transition matrix needs to be learned simultaneously, the space efficiency of CT offers us an advantage for model learning. For tasks that involve very long sequences, we can limit the depth D of the CT for space and time efficiency.

4. DATASET AND PROBLEM ANALYSIS

4.1 Dataset Description

In this paper, we work with recommending discussion forum threads to MOOC students. A forum thread can be updated frequently and it contains multiple posts and comments within the topic. As we mentioned before that the challenge is adapting to drifting user preferences and evolving forum threads as a course progresses. For the experiments elaborated in the following section, we use forum viewing data from three courses offered by École polytechnique fédérale de Lausanne on Coursera. These three courses include the first offering of "Digital Signal Processing", the third offering of "Functional Program Design in Scala", and the first offering of "'Reactive Programming'. They are referred to *Course 1*, *Course 2* and *Course 3*. Some discussion forum statistics for the three courses are given in Table 1. From the number of forum participants, forum threads, and thread views, we can see that the course scale increase from *Course 1* to *Course 3*. A student on MOOCs often accesses course forums many times during the span of a MOOC. Each time the threads she views are tracked as one *visit session* by the web browser. The total number of visit sessions and the average session lengths for three courses are presented in Table 1. The length of a session is the number of threads she viewed within a visit session. The thread viewing sequences corresponding to these regular visit sessions are called *separated* sequences in our later experiments and they treat threads in one visit session as one sequence. Models built using separated sequences try to catch short-term patterns within one visit session and we do not differentiate the patterns from different students. Another setting, called *combined* sequences, concatenates all of a student's visit sessions into one longer sequence so that models built using combined sequences try to learn long-term patterns across students. The average length of combined sequences is the average session length times the average number of sessions per student. From *Course 1* to *Course 3*, average lengths for separated and combined sequences both increase.

	Course 1	*Course 2*	*Course 3*
# of forum participants	5,399	12,384	13,914
# of forum threads	1,116	1,646	2,404
# of thread views	130,093	379,456	777,304
# of sessions	19,892	40,764	30,082
avg. session length	6.5	9	25.8
avg. # of sessions per student	3.7	3.3	2.2

Table 1: Course forum statistics for three datasets.

Another important issue that we can discover from the statistics is that thread viewing data available for sequential recommendation is very sparse. For example in *Course 1*, the average session length is 6.5 and the number of threads is around 1116. Then the complete space to be explored will be $1116^{6.5}$, which is much larger than the size of observations (130,093 thread views). The similar data sparsity issue is even more severe in the other two datasets.

4.2 Forum Thread View Pattern

Next, we study the thread viewing pattern which highlights the significance of adaptation issues for thread recommendation. Figure 3 illustrates the distribution of thread views against *freshness* for three courses. The freshness of an item is defined as the relative creation order of all items that have been created so far. For example, when a student views a thread t_m which is the m-th thread created in the currently existing pool of n threads, then *freshness* of t_m is defined as:

$$freshness = \frac{m}{n} \tag{6}$$

We can see from Figure 3 that there is a sharp trend that the new forum threads are viewed much more frequently than the old ones for all three courses. It is mainly due to the fact that fresh threads are closely relevant to the current course progress. Moreover, fresh threads can also supersede the contents in some old ones to be viewed. This tendency to view fresh items leads to drifting user preferences. Such drifting preferences, coupled with the evolving nature of forum contents, requires recommendations adaptive to drifting or recent preferences.

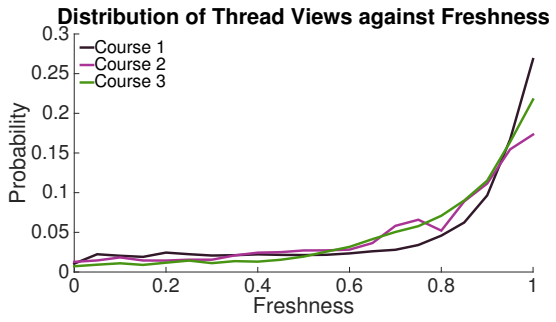


Figure 3: Thread viewing activities against freshness

A further investigation through those views on old threads leads us to a classification of threads into two categories: *general threads* and *specific threads*. Some titles of the general and specific threads are listed in Table 2. We could see the clear difference between these two classes of threads as the general ones corresponds to broad topics and specific ones are related to detailed course contents or exercises. We also found that only a very small part of the old threads are still rather active to be viewed and they are mostly general ones. Different from general threads, specific threads that subject to a fine timeliness are viewed very few times after they get old. In general, sequential patterns are observed more often within specific threads as some specific follow-up threads might be related and useful to the one that you are viewing. So the patterns learned could be used to guide your forum browsing process. On the contrary, sequential patterns on general threads are relatively random and imperceptible.

General Threads	**Specific Threads**
"Using GNU Octave"	"Homework Day 1 / Question 9"
"Any one from INDIA??"	"Quiz for module 4.2"
"Where is everyone from?	"quiz -1 Question 04"
"Numerical Examples in pdf"	'Homework 3, Question 11"
"How to get a certificate"	"Week 1: Q10 GEMA problem"

Table 2: Sample thread titles of general and specific threads.

5. RESULTS AND EVALUATION

In this section, we compare the proposed CT method against various baseline methods in both non-sequential and sequential settings. The results show that the CT recommender performs better than other methods under different setting for all three MOOCs considered. Through the adaptation analysis, we validate our hypothesis that the superior performance of CT recommender comes from the adaptation power to drifting preferences and trendy patterns in the domain. In the end, a regularization technique for CT, called partial context matching (PCT), is introduced. It is demonstrated that PCT helps better generalize among sequence patterns and further boost performance.

5.1 Baseline Methods

5.1.1 Non-sequential Methods

Matrix factorization methods proposed by [23, 22] are the state-of-the-art for MOOCs course content recommendation. Besides the user-based MF given in [23], we also consider item-based MF that generates recommendations based on the similarity of the latent item features learned from standard MF. In our case, each entry in the user-item matrix of MF contains the number of times a student views a thread. We also test a version where the matrix had a 1 for any number of views, but the performance was not as good, so the development of this version was not taken any further. MF models considered here are updated periodically (week-by-week). To enable a fair comparison against non-sequential matrix factorization techniques, we implemented versions where the CT model is updated at fixed time intervals, equal to those of the MF models. In the "One-shot CT" version, we compute the CT recommendations for each user based on the data available at the time of the model update, and the user then receives these same recommendations at every future time step until the next update. This mirrors the conditions of user-based MF. To compare with item-based MF, the "Slow-update CT" version updates the recommendations, but not the model, at each time point based on the sequential forum viewing information available at that time.

5.1.2 Sequential Methods

Sequential methods update model parameters and recommendations continuously as items are consumed. The first two simple methods are based on the observation and heuristic that fresh threads are viewed much frequently than old ones. *Fresh_1* recommends the last 5 *updated* threads, and *Fresh_2* recommends the last 5 *created* threads. Another baseline method, referred as *Popular*, recommends the top 5 threads among the last 100 threads viewed before the current one. We also consider an online version of MF [10] that

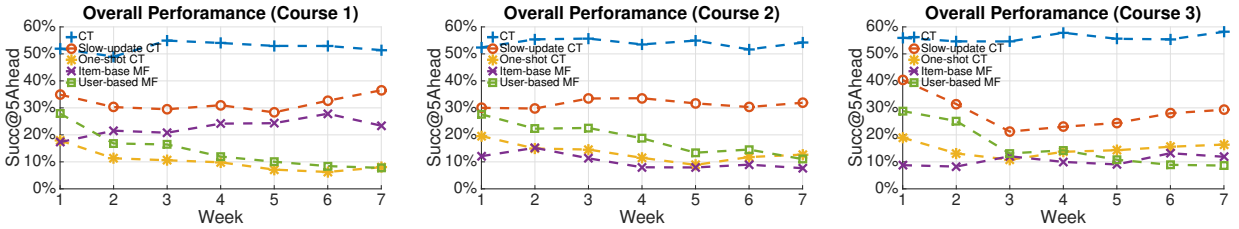


Figure 4: Overall performance comparison of CT and non-sequential methods

is currently the state-of-the-art sequential recommendation method, referred to "online-MF", in which the corresponding latent factor of the item i and user u are updated when a new observation R_{ui} arrive. The model optimization is implemented based on element-wise Alternating Least Squares. The number of latent factors is tuned to be 15, 20, 25 for three datasets, and the regularization parameter is set as 0.01. Moreover, the weight of a new observation is the same as old ones during optimization for achieving the best performance. Furthermore, the proposed CT recommender refers to the full context tree algorithm with a continuously updated model.

5.2 Performance and Adaptation Analysis

5.2.1 Evaluation Metrics

In our case, all methods recommend top-5 threads each time. Two evaluation metrics are adopted in the following experiments:

- **Succ@5**: the mean average precision (MAP) of predicting the immediately next thread view in a sequence.
- **Succ@5Ahead**: the MAP of predicting the future thread views within a sequence. In this case, a recommendation is successful even if it is viewed later in a sequence.

5.2.2 Comparison of Non-sequential Methods

Figure 4 shows the performance comparison between different versions of methods based on MF and CT on three datasets. "CT" is the sequential method with a continuously updated model, and all other methods Figure 4 are non-sequential versions. *Combined* sequences are used for the CT methods here to have a parallel comparison against MF. We found that a small value of the depth limit of the CTs hurts performances, yet a very large depth limit does not increase performance at the cost of computation and memory. Through experiments, we tune depths empirically and set them as 15, 20, 30 for three datasets.

Among non-sequential methods, one-shot CT and user-based MF perform the worst for all three courses, which means that recommending the same content for the next week without any sequence consideration is ineffective. Slow-update CT performs consistently the best among non-sequential methods, and it proves that adapting recommendations through context tree helps boost performance although the model itself is not updated continuously. Compared to slow-update CT, item-based MF performs much worse. They both update model parameters periodically and the recommendations are adjusted given the current observation. However, using the contextual information within a sequence and the corresponding prediction experts of slow-update CT are much more powerful than just using latent item features of item-base MF. Moreover, we can clearly see that the normal CT with continuous update outperforms all other non-sequential methods by a large margin for three datasets. It means that drifting preferences need to be followed though continuous and adaptive model update, so sequential methods are better choices. Next, we focus on sequential methods, and we validate our hypothesis that the CT model has superior performances because it better handles drifting user preferences.

5.2.3 Comparison of Sequential Methods

The results presented in Table 3 show the performance of the full CT recommender compared with other sequential baseline methods under different settings and evaluation metrics. Each result tuple contains the performance on the three datasets. We also consider a *tail performance* metric, referred to *personalized* evaluation, where the most popular threads (20, 30, and 40 for three courses) are excluded from recommendations. The depth limits of CTs using *separated* sequences are set to 8, 10, and 15 for three courses.

We notice that the online-MF method, with continuous model update, performs much worse compared with the CT recommender for all three datasets. This result shows that matrix factorization, which is based on interpolation over the user-item matrix, is not sensitive enough to rapidly drifting preferences with limited observations. The performances of two versions of the *Fresh* recommender are comparable with online-MF, and *Fresh_1* even outperforms online-MF in many cases, especially for **Succ@5Ahead**. It means that simply recommending fresh items even does a better job than online-MF for this recommendation task with drifting preferences. We can see that the CT recommender outperforms all other sequential methods under various settings, except for using non-personalized Succ@5Ahead for *Course 2*. The *Popular* recommender is indeed a very strong contender when using non-personalized evaluation since there is a bias that students can click a "top threads" tag from user interface to view popular threads which are similar to the ones given by *Popular* recommender. From the educational perspective, the setting using separated sequences and personalized evaluation is the most interesting as it reflects shot-term visiting patterns within a session over those specific and less popular forum threads. We could see from the upper right part of Table 3 that the CT recommender outperforms all other methods by a large margin under this setting.

	Non-personalized		*Personalized*	
	Succ@5	Succ@5Ahead	Succ@5	Succ@5Ahead
Separated Sequences				
CT	[**25**, **23**, **21**]%	[**48**, 53, **52**]%	[**19**, **14**, **16**]%	[**41**, **37**, **42**]%
online-MF	[15, 12, 8]%	[33, 29, 23]%	[10, 7 ,6]%	[27, 25, 20]%
Popular	[15, 20, 16]%	[40, **61**, 51]%	[9, 8 ,8]%	[34, 31, 36]%
Fresh_1	[12, 14, 10]%	[37, 43, 41]%	[10, 10, 8]%	[33, 31, 37]%
Fresh_2	[9, 8, 6]%	[31, 31, 29]%	[8, 7, 6]%	[30, 30, 28]%
Combined Sequences				
CT	[**21**, **20**, **20**]%	[**55**, 55, **56**]%	[**16**, **13**, **14**]%	[**46**, **39**, **46**]%
online-MF	[9, 8, 7]%	[34, 27, 23]%	[7,6,6]%	[29, 24, 20]%
Popular	[13, 14, 14]%	[52, **62**, 58]%	[9, 8, 7]%	[45, 36, 43]%
Fresh_1	[10, 12, 9]%	[48, 44, 44]%	[8, 9, 8]%	[44, 34, 42]%
Fresh_2	[7, 6, 6]%	[43, 34, 32]%	[6, 6, 6]%	[42, 32, 31]%

Table 3: Performance comparison of sequential methods

5.2.4 Adaptation Comparison

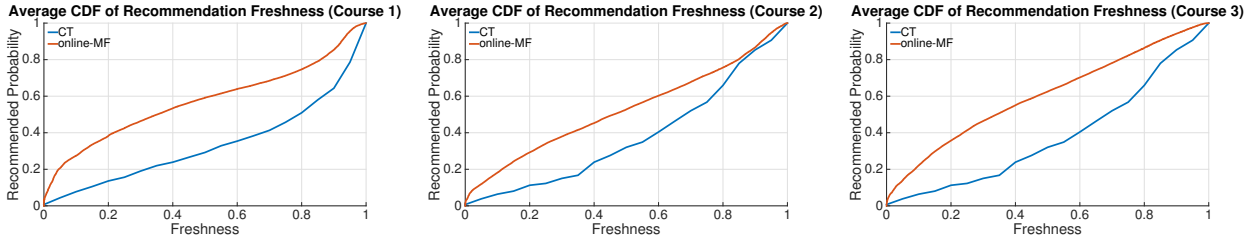


Figure 5: Distribution of recommendation freshness of CT and online-MF

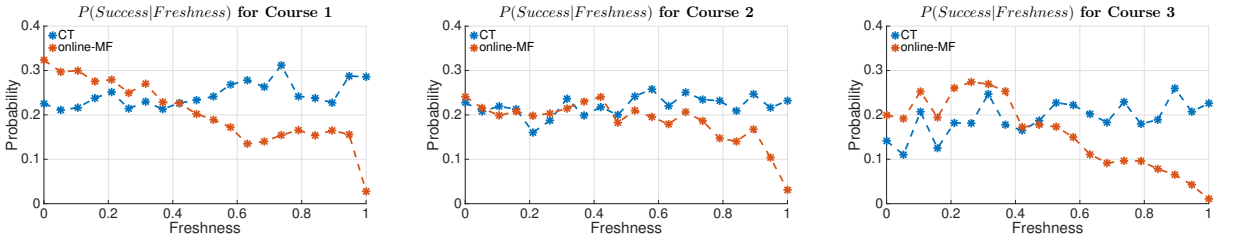


Figure 6: Conditional success rate of CT and online-MF

After seeing the superior performance of the CT recommender, we move to an insight analysis of the results. To be specific, we compare CT and online-MF in terms of their adaptation capabilities to new items. Figure 5 illustrates the cumulative density function (CDF) of the threads recommended by different methods against thread freshness. We can see that the CDFs of CT increase sharply when thread freshness increases, which means that the probability of recommending fresh items is high compared to online-MF. In other words, CT recommends more fresh items than online-MF. As we mentioned before that a large portion of fresh threads are specific ones, instead of general ones, so CT recommends more specific and trendy threads to students while methods based on matrix factorization recommend more popular and general threads.

Other than the quantity of recommending fresh and specific threads, the quality is crucial as well. Figure 6 shows the conditional success rate $P(Success|Freshness)$ across different degrees of freshness for three courses. $P(Success|Freshness)$ is defined as the fraction of the items successfully recommended given the item freshness. For instance, if an item with freshness 0.5 is viewed 100 times throughout a course, then $P(Success|Freshness = 0.5) = 0.25$ means it is among the top 5 recommended items 25 times. As the freshness increases, the conditional success rate of online-MF drops speedily while the CT method keeps a solid and stable performance. It is significant that CT outperforms online-MF by a large margin when freshness is high, in other words, it is particularly strong for recommending fresh items. Fresh items are often not popular in terms of the total number of views at the time point of recommendation. So identifying fresh items accurately implies a strong adaptation power to new and evolving forum visiting patterns. The analysis above validates our hypothesis that the CT recommender can adapt well to drifting user preferences. Another conclusion drawn from Figure 6 is that the performance of CT is as good as online-MF for items with low freshness. This is because that the context organization and context matching mechanism help old items to be identifiable though old contexts. To conclude, CT is flexible at combining old knowledge and new knowledge so that it performances well for items with various freshness, especially for fresh ones with drifting preferences.

5.3 Partial Context Matching (PCT)

At last, we introduce another technique, built on top of the standard CT, to generalize to new sequence patterns and further boost the recommendation performance. The standard CT recommender adopts a complete context matching mechanism to identify active experts for a sequence $\mathbf{s}$. That is, active experts of $\mathbf{s}$ come exactly from the set of suffixes of $\mathbf{s}$. We design a partial context matching (PCT) mechanism where active experts of a sequence are not constrained by exact suffixes, yet they can be those very similar ones. Two reasons bring us to design the PCT mechanism for context tree learning. First, PCT mechanism is a way of adding regularization. Sequential item consumption process does not have to follow exactly the same order, and slightly different sequences are also relevant for both model learning and recommendation generation. Second, the data sparsity issue we discussed before for sequential recommendation setting can be solved to some extent by considering similar contexts for learning model experts. The way PCT does aims to activate more experts to train the model, and to generate recommendations from a mixture of similar contexts.

We will focus on a *skip* operation that we add on top of the standard CT recommender. Some complex operations, like swapping item orders, are also tested, but they do not generate better performance. For a sequence $\langle s_p, \ldots, s_1 \rangle$ with length p, the skip operation generates p candidate partially matched contexts that skip one s_k for $k \in [1 \ldots p]$. All the contexts on the paths from root to partially matched contexts are activated. For example, the path to context $\langle n_2, n_1 \rangle$ can be activated from the context $\langle n_2, n_3, n_1 \rangle$ by the skipping n_3. However, for each partially matched context, there may not exist a fully matched path in the current context tree. In this case, for each partially matched context, we identify the longest path that corresponds it with length q. If q/p is larger than some threshold t, we update experts on this paths and use them to generate recommendations for the current observation. Predictions from multiple paths are combined by averaging the probabilities.

	Success@5	Success@5Ahead	Ratio
PCT-0.5	[+0.4, +0.6, +0.2]%	[+0.8, +0.9, +0.4]%	[4.9, 4.5, 3.3]
PCT-0.6	[+0.5, +0.8, +0.3]%	[+1.1, +1.3, +0.5]%	[4.4, 4.1, 2.9]
PCT-0.7	[+0.7, +0.9, +0.5]%	[+1.6, +1.9, +0.7]%	[3.7, 3.2, 2.5]
PCT-0.8	[+0.8, +1.1, +0.6]%	[+1.9, +2.4, +1.0]%	[3.2, 2.9, 2.1]
PCT-0.9	[+1.0, +1.4, +0.7]%	[+2.0, +2.7, +1.3]%	[2.4, 2.2, 1.4]

Table 4: Performance comparison of PCT against CT for three courses

Table 4 shows the performance of applying PCT for both model update and recommendation with threshold t (PCT-t). Results are compared with the full CT recommender with separated sequences and non-personalized evaluation. For cases where the threshold is smaller than 0.5, we sometimes obtain negative results since partially matched contexts are too short to be relevant. The "Ratio" column is the ratio of the number of updated paths in PCT compared with standard CT. We can see that PCT updates more paths and it offers us consistent performance boosts at the cost of computation.

6. CONCLUSION AND FUTURE WORK

In this paper, we formulate the MOOC forum recommendation problem as a sequential decision problem. Through experimental analysis, both performance boost and adaptation to drifting preferences are achieved using a new method called context tree. Furthermore, a partial context matching mechanism is studied to allow a mixture of different but similar paths. As a future work, exploratory algorithms are interesting to be tried. As exploring all options for all contexts are not feasible, we consider to explore only those top options from similar contexts. Deploying the CT recommender in some MOOCs for online evaluation would be precious to obtain more realistic evaluation.

7. REFERENCES

[1] J. Abernethy, K. Canini, J. Langford, and A. Simma. Online collaborative filtering. *University of California at Berkeley, Tech. Rep*, 2007.

[2] R. Begleiter, R. El-Yaniv, and G. Yona. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, pages 385–421, 2004.

[3] R. M. Bell, Y. Koren, and C. Volinsky. The Bellkor 2008 solution to the Netflix prize. *Statistics Research Department at AT&T Research*, 2008.

[4] G. Bonnin, A. Brun, and A. Boyer. A low-order Markov model integrating long-distance histories for collaborative recommender systems. In *International Conference on Intelligent User Interfaces*, pages 57–66. ACM, 2009.

[5] C. Dimitrakakis. Bayesian variable order Markov models. In *International Conference on Artificial Intelligence and Statistics*, pages 161–168, 2010.

[6] Y. Ding and X. Li. Time weight collaborative filtering. In *ACM International Conference on Information and Knowledge Management*, pages 485–492. ACM, 2005.

[7] Y. Ding, X. Li, and M. E. Orlowska. Recency-based collaborative filtering. In *Australasian Database Conference*, pages 99–107. Australian Computer Society, Inc., 2006.

[8] F. Garcin, C. Dimitrakakis, and B. Faltings. Personalized news recommendation with context trees. In *ACM Conference on Recommender Systems*, pages 105–112. ACM, 2013.

[9] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber. Offline and online evaluation of news recommender systems at swissinfo.ch. In *ACM Conference on Recommender Systems*, pages 169–176. ACM, 2014.

[10] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua. Fast matrix factorization for online recommendation with implicit feedback. In *International ACM Conference on Research and Development in Information Retrieval*, volume 16, 2016.

[11] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[12] S. S. Kozat, A. C. Singer, and G. C. Zeitler. Universal piecewise linear prediction via context trees. *IEEE Transactions on Signal Processing*, 55(7):3730–3745, 2007.

[13] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web*, pages 661–670. ACM, 2010.

[14] E. Liebman, M. Saar-Tsechansky, and P. Stone. DJ-MC: A reinforcement-learning agent for music playlist recommendation. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 591–599. IFAAMAS, 2015.

[15] N. N. Liu, M. Zhao, E. Xiang, and Q. Yang. Online evolutionary collaborative filtering. In *ACM Conference on Recommender Systems*, pages 95–102. ACM, 2010.

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

[17] G. Shani, R. I. Brafman, and D. Heckerman. An MDP-based recommender system. In *Conference on Uncertainty in Artificial Intelligence*, pages 453–460. Morgan Kaufmann Publishers Inc., 2002.

[18] A. Srivihok and P. Sukonmanee. E-commerce intelligent agent: personalization travel support agent using Q-Learning. In *International Conference on Electronic Commerce*, pages 287–292. ACM, 2005.

[19] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4, 2009.

[20] X. Wang, Y. Wang, D. Hsu, and Y. Wang. Exploration in interactive personalized music recommendation: a reinforcement learning approach. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11(1):7, 2014.

[21] F. M. Willems, Y. M. Shtarkov, and T. J. Tjalkens. The context-tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, 41(3):653–664, 1995.

[22] D. Yang, D. Adamson, and C. P. Rosé. Question recommendation with constraints for massive open online courses. In *ACM Conference on Recommender Systems*, pages 49–56. ACM, 2014.

[23] D. Yang, M. Piergallini, I. Howley, and C. Rose. Forum thread recommendation for massive open online courses. In *Educational Data Mining*, 2014.

[24] A. Zimdars, D. M. Chickering, and C. Meek. Using temporal data for making recommendations. In *Conference on Uncertainty in Artificial Intelligence*, pages 580–588. Morgan Kaufmann Publishers Inc., 2001.

Analysis of problem-solving behavior in open-ended scientific-discovery game challenges

Aaron Bauer
awb@cs.washington.edu

Jeff Flatten
jflat06@cs.washington.edu

Zoran Popović
zoran@cs.washington.edu

Center for Game Science, Computer Science and Engineering
University of Washington
Seattle, WA 98195, USA

ABSTRACT

Problem-solving skills in creative, open-ended domains are both important and little understood. These domains are generally ill-structured, have extremely large exploration spaces, and require high levels of specialized skill in order to produce quality solutions. We investigate problem-solving behavior in one such domain, the scientific-discovery game *Foldit*. Our goal is to discover differentiating patterns and understand what distinguishes high and low levels of problem-solving skill. To address the challenges posed by the scale, complexity, and ill-structuredness of *Foldit* solver behavior data, we devise an iterative visualization-based methodology and use this methodology to design a concise, meaning-rich visualization of the problem-solving process in *Foldit*. We use this visualization to identify key patterns in problem-solving approaches, and report how these patterns distinguish high-performing solvers in this domain.

Keywords

Problem Solving; Scientific-Discovery Games; Visualization

1. INTRODUCTION

As efforts in scalable online education expand, interest continues to increase in moving beyond small, highly constrained tasks, such as multiple choice or short answer questions, and incorporating creative, open-ended activities [7, 14]. Existing research supports this move, showing that problem-based learning can enhance students' problem-solving and metacognitive skills [11]. Scaling such activities poses significant challenges, however, in terms of both assessment and feedback. It will be vital to devise scalable techniques not only to assess students' final products, but also to understand their progress through complex and heterogeneous problem-solving spaces. These techniques will apply to a broad range of education settings, from purely online programs like Udacity's Nanodegrees to more traditional settings where new standards like the Common Core emphasize strategic problem solving.

A growing body of work has found that educational and serious games are fertile ground for assessing students' capabilities and problem-solving skills [6, 10]. Our work continues this general line of inquiry by examining creative, problem-solving behavior among players in the scientific-discovery game *Foldit*. By modeling the functions of proteins, the workhorses of living cells, *Foldit* challenges players, hereafter referred to as solvers, to resolve the shape of proteins as a 3D puzzle. These puzzles are completely open and often under-specified, making it a highly suitable setting in which to gain insight into student progress through complex solution spaces. In the *Foldit* scientific-discovery community, the focus is on developing people from novices to experts that are eventually capable of solving protein structure problems that are currently unsolved by the scientific community. In fact, solutions produced in *Foldit* have led to three results published in Nature [3, 5, 16]. *Foldit* is an attractive learning space domain because its solvers are capable of contributing to state-of-the-art biochemistry results, and the vast majority of best performing solvers had no exposure to biochemistry prior to joining *Foldit* community. Hence, solver behavior in *Foldit* represents development of highly effective problem-solving in an open-ended domain over long time horizons. In this work, we identify six strategic patterns employed by *Foldit* solvers and show how these patterns differentiate between successful and less successful solvers. These patterns cover instances where solvers investigate multiple hypotheses, explore more greedily or more inquisitively, try to escape local optima, and make structured use of the manual or automated tools available in *Foldit*.

The aspects of the *Foldit* environment that make it an attractive setting in which to study problem solving also present significant challenges. Problems in *Foldit* share many of the properties Jonassen attributes to *design problems*, which they describe as "among the most complex and ill-structured kinds of problems that are encountered in practice" [13]. These properties include a *vague goal with few constraints* (in *Foldit*, the goal is often entirely open-ended: find a good configuration of the protein), *answers that are neither right or wrong, only better or worse*, and *limited feedback* (in *Foldit*, real-time feedback and solution evaluation are limited to a single numerical score corresponding to the protein's current energy state, and solvers frequently must progress through many low-scoring states to reach a good configuration; more nuanced feedback from biochemists is sometimes available, but on a timescale of weeks). The ill-structured nature of problems posed in *Foldit* necessarily deprives us of the structures, such as clear goal states and straightforward relationships between intermediate states and goal states, that typically form the basis of existing detailed and quantitative analyses of problem-solving behavior.

The size and complexity of *Foldit*'s problem space presents another major challenge. Even though the logs of solver interactions consist only of regular snapshots of a solver's current solution (along with attendant metadata), the record of a single solver's performance on a given problem frequently consists of thousands of such snapshots (which in turn are just a sparse sampling of the actual solving process). Furthermore, the nature of the solution state, the configuration of hundreds of components in continuous three-dimensional space, renders collapsing the state space by directly comparing solution states impractical. Compounding the size of the problem space is the complexity of the actions available to *Foldit* solvers. In addition to manual manipulation of the protein configuration, solvers can invoke various low-level automated optimization routines (some

of which run until the solver terminates them) and place different kinds of constraints on the protein configuration (*rubber bands* in *Foldit* parlance) that restrict its modification in a variety of ways. Solvers can also deploy many of these tools programmatically via Lua scripts called *recipes*. Taken together these challenges of ill-structuredness, size, and complexity threaten to make analysis of high-level problem-solving behavior in *Foldit* intractable.

To overcome these obstacles, we devise a visualization-based methodology capable of producing tractable representations of *Foldit* solvers' problem-solving behavior while maintaining the key encodings necessary for analysis of high-level strategic behavior. A process of iterative summarization forms the core of this methodology, and ensures that the transformations applied to the raw data do not elide structures potentially relevant to understanding solvers' unique strategic behavior. Using this methodology, we examine solver activity logs from 11 *Foldit* puzzles, representing 970 distinct solvers and nearly 3 million solution snapshots. Leveraging metadata present in the solution snapshots, we represent solving behavior as a tree, and apply our methodology to visualize a summarized tree showing where they branched off to investigate multiple hypotheses, how they employed some of the automated tools available to them, and other salient problem-solving behavior. We use these depictions to determine key distinguishing features of this exploration process. We subsequently use these features to better understand the patterns of expert-level problem solving.

Our work focuses on the following research questions: (1) how can we visually represent an open-ended exploration towards a high-quality solution in a large, ill-structured problem space? (2) what are the key patterns of problem-solving behavior exhibited by individuals?, and (3) what are the key differences along these patterns between high-performing and lower-performing solvers in an open-ended domain like *Foldit*? In addressing these questions we find that high-performing solvers explore the solution space more broadly. In particular, they pursue more hypotheses and actively avoid getting stuck in local minima. We also found that both high- and lower-performing solvers have similar proportion of manual and automated tool actions, indicating that better performance on open-ended challenges stems from the quality of the action intermixing rather than aggregate quantity.

2. RELATED WORK

While automated grading has mostly been explored for well-specified tasks where the correct answer has a straightforward and concise description, some previous work has developed techniques for more complex activities. Some achieve scalability through a crowd-sourcing framework such as Udacity's system for hiring external experts as project reviewers [14]. Other work has demonstrated automated approaches that leverage machine learning to enable scalable grading of more complex assignments. For example, Geigle et al. describe an application of online active learning to minimize the training set a human grader must produce [7] when automatically grading an assignment where students must analyze medical cases. Our work does not focus on grading problem-solving behavior, but instead approaches the issue of scalability at a more fundamental level: understanding fine-grained problem-solving strategies and how they contribute to success in an open-ended domain.

A robust body of prior work has addressed the challenge of both visualizing and gleaning insight from player activity in educational and serious games. Andersen et al. developed Playtracer, a general method for visualizing players' progress through a game's state space when a spatial relationship between the player and the virtual environment is not available [1]. Wallner and Kriglstein provide a thorough review of visualization-based analysis of gameplay data [21]. Prior work has analyzed gameplay data without visualization as well. Falakmasir et al. propose a data analysis pipeline for modeling player behavior in educational games. This system can produce a simple, interpretable model of in-game actions that can predict learning outcomes [6]. Our work differs in its aims from this prior work. We do not seek to develop a general visualization technique, but instead to design and leverage a domain-specific visualization to analyze problem-solving behavior. We are also not predicting player behavior, nor modeling players in terms of low-level actions, but rather identifying higher-level strategy use.

The work most similar to ours is that which focuses on problem-solving behavior, including both the long-running efforts in educational psychology to develop general theories and more recent work data-driven on understanding the problem-solving process. Our formulation of solving behavior in *Foldit* as a search through a problem space follows from classic information-processing theories of problem solving (e.g., [9, 19]). Gick reviews research on both problem-solving strategies and the differences in strategy use between experts and novices [8]. Our work complements the existing literature by focusing on understanding problem solving in the little-studied domain of scientific-discovery games, and on the ill-structured problems present in *Foldit*. Our findings on the differences in strategy use between high- and lower-performing solvers in *Foldit* are consistent with the consensus in the literature that expert's knowledge allows them to effectively use strategies that are poorly or infrequently used by less-skilled solvers. We also contribute a granular understanding of the specific strategies and differences at work in the *Foldit* domain.

Significant recent work has investigated problem-solving behavior in educational games and intelligent tutoring systems using a variety of techniques. Tóth et al. used clustering to characterize problem-solving behavior on tasks related to understanding a system of linear structural equations. The clusters distinguished between students that used a *vary-one-thing-at-a-time* strategy (both more and less efficiently) and those that used other strategies [20]. Through a combination of automated detectors, path analysis, and classroom studies, Rowe et al. investigated the relationship between a set of six strategic moves in a Newtonian physics simulation game and performance on pre- and post-assessments. They found that the use of some moves mediated the relationship between prior achievement and post scores [18]. Eagle et al. discuss several applications of using *interaction networks* to visualize and categorize problem-solving behavior in education games and intelligent tutoring systems. These networks offer insight for hint generation and a flexible method for visualizing student work in rule-using problem solving environments [4] . Using decision trees to build separate models for optimal and non-optimal student performance, Malkiewich et al. gained insight into how learning environments can encourage elegant problem solving [17]. Our primary contribution is to extend analysis of problem-solving behavior to a more complex and open-ended domain that those studied in similar previous work. The size and complexity of *Foldit*'s problem space, the volume of data necessary to capture exploration in this space, and the ill-structured nature of the *Foldit* problems all pose unique challenges. We devise a visualization-based methodology focused on iterative summarization, and successfully apply it to identify key problem-solving patterns exhibited by *Foldit* solvers.

3. FOLDIT

Foldit is a scientific-discovery game that crowdsources protein folding. It presents solvers with a 3D representation of a protein and tasks them with manipulating it into the lowest energy configuration. Each protein posed to the solvers is called a puzzle. Solvers' solutions to each puzzle are scored according to their energy configuration, and solvers compete to produce the highest scoring results.

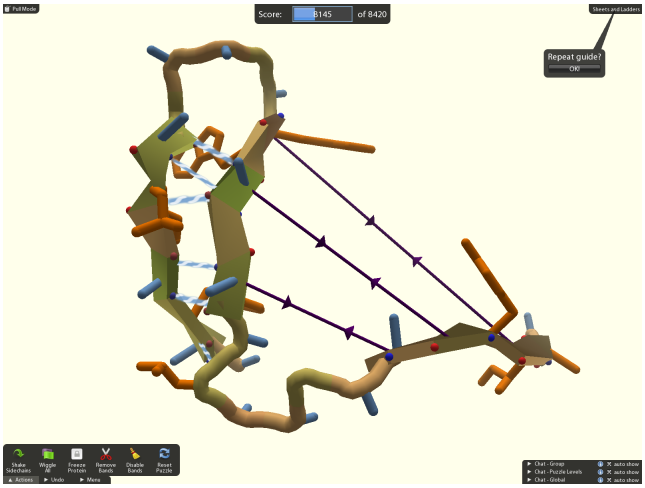


Figure 1: The *Foldit* interface. *Foldit* solvers use a variety of tools to interactively reshape proteins. In this figure, a solver uses rubber bands to pull together two sheets, long flat regions of the protein.

Solvers have many tools at their disposal when solving *Foldit* puzzles. They can manipulate and constrain the structure in various ways, employ low-level automated optimization (e.g., a *wiggle* tool makes small, rapid, local adjustments to try and improve the score), and trigger solver-created automated scripts called *recipes* that can programmatically use the other tools. There is, however, a subset of the basic actions that cannot be used by recipes. We will call these *manual-only actions*. Previous work analyzing solver behavior in *Foldit* has focused primarily on recipe use and dissemination [2] and recipe authoring [15].

Foldit has several different types of puzzles for solvers to solve. In this work, we focus on the most common type of puzzle, *prediction* puzzles. These are puzzles in which biochemists know the amino acids that compose the protein in question, but do not know how the particular protein folds up in 3D space. This is in contrast to *design* puzzles in which solvers insert and delete which amino acids compose the protein to satisfy a variety of scientific goals, including designing new materials and targeting problematic molecules in diseases. We focus on prediction puzzles in this work to simplify our analysis by having a consistent objective (i.e., maximize score) across the problem-solving behavior we analyze.

4. METHODOLOGY

Prior work has demonstrated the power of visualization to support understanding of problem-solving behavior (e.g., [12]). Hence, we devise a methodology capable of producing concise, meaning-rich visualizations of the problem-solving process in *Foldit*, and then leverage these visualizations to identify key patterns of solver behavior. We are specifically interested in how solvers navigate from a puzzle's start state to a high-quality solution, what states they pass through in between, and what other avenues they explored. Since solving a *Foldit* puzzle can be represented as a directed search through a problem space, the clear encoding of parent-child relationships between nodes offered by a tree make it well-suited for visualizing these aspects of the solving process.

The scale of the *Foldit* data necessitates significant transformation of the raw data in order to render concise visualizations. Without any transformation, meaningful patterns are overwhelmed by sparse, repetitive data and would be far more challenging to identify. While there are many existing techniques for large-scale tree visualization, we find clear benefits to developing a visualization tailored to the *Foldit* domain. Specifically, preserving the semantics of our visual encoding is crucial for allowing us to connect patterns in the visualization to concrete strategic behavior in *Foldit*. To accomplish this, the process by which concise visualization are constructed must be carefully designed to maintain these links. Hence, we devise a design methodology focused on *iterative summarization*.

This process begins by visualizing the raw data. This is followed by iteratively building and refining a set of transformations to summarize the raw data while preserving meaning. The design of these transformations should be guided by frequently occurring structures. That is, those structures that the transformations can condense without eliding structures corresponding to unique strategic behavior. In parallel to this iterative design, a set of visual encodings are developed to represent the solving process as richly as possible. Key to this entire process is frequent consultation with domain experts, in our case experts on *Foldit* and its community. By applying this iterative methodology for several cycles, we designed a domain-specific visualization that we use to identify patterns of strategic behavior among *Foldit* solvers. We follow up on these patterns with computational investigation, and quantify their application by high- and lower-performing solvers.

4.1 Data

For our analysis, we selected 11 prediction puzzles spanning the range of time for which the necessary data is available. Though *Foldit* has been in continuous use since 2010, the data necessary to track a solver's progress through the problem space has only been collected since mid-2015. Our chosen dataset represents 970 unique solvers and nearly 3 million solution snapshots. These 11 puzzles are just a small subset of the available *Foldit* data. We chose a subset of similar puzzles (i.e., a subtype of relatively less complex prediction puzzles) in order to make common solving-behavior patterns easier to identify. The size of the subset was also guided by practical constraints, as each puzzle constitutes a large amount of data (20-60 GB for the data from all players on a single puzzle).

The data logged by *Foldit* primarily consists of snapshots of solver solutions as they play, stored as text files using the Protein Data Bank (pdb) format. These snapshots include the current protein pose, a timestamp, the solution's score, the number of times the solver has invoked each action and recipe, and a record of the intermediate states that led up to the solution at the time of the snapshot. This record, or *solution history*, is a list of unique identifiers each corresponding to a previous solution state. This list is extended every time the solver undoes an action or reloads a previous solution. Hence, by comparing the histories of two snapshots from the same solver, we can answer questions about their relationship (e.g., does one snapshot represent the predecessor of another; where did two related snapshots diverge). The key relationship for the purposes of this analysis is the direct parent-child relationship, which we use to generate trees that represent a solver's solving process.

4.2 Visualizing Solution Trees

We applied our methodology to our chosen subset of *Foldit* data to design a visualization of an individual's problem-solving process as a *solution tree*. Several key principles guided this design. First, since our goal is to discover key patterns, the visualization needs to highlight distinctly different strategies and approaches. These differences cannot be buried amidst enormous structures, nor destroyed by graph transformations. Second, the visualization must depict the closeness of each step to the ultimate solution in both time and quality to give a sense of the solver's progression. Third, the solver's use of automation in the form of recipes should be apparent since the use of automation is an important part of *Foldit*.

The fundamental organization of the visualization is that each node corresponds to a solution state encountered while solving. Using the solution history present in the logged snapshots of solver solutions, we establish parent-child relationships between solutions. If solution β is a child of solution α, it indicates that β was generated when the solver performed actions on α. One crucial limitation, however, is that a snapshot of the solver's current solution is captured far less often (only once every two minutes) than the solver takes actions. This means that our data is sparsely distributed along a solution's history going back to the puzzle's starting state. Hence, when naively constructing the tree from the logged solution histories, it ends up dominated by vast quantities of nodes with no associated data.

We address this issue by performing summarization on the solution trees, condensing them into concise representations amenable to analysis for important features. This summarization takes place in two stages. The first stage trims out nodes that (1) do not have corresponding data and (2) have zero children. This eliminates large numbers of leaf nodes that we are unable to reason about given that we lack the corresponding data. This stage also combines sequences of nodes each with only one child into a single node. For the median tree, this stage reduced the number of nodes by an order of magnitude from over 12,000 nodes to about 1,600.

The second stage consists of four phases, each informed by our observations of common patterns in trees produced by the first stage that would benefit from summarization. The first phase, called *prune*, focuses on simplifying uninteresting branches. We observed many of the branches preserved by the first stage were small, with at most three children, and only continued the tree from one of those children. Prune removes the leaf children of these branches from the tree. *Collapse*, the second phase, transforms each of the sequences of single-child nodes left behind after prune into single nodes. The third phase, *condense*, targets another common pattern where a sequence of branches feed into each other, with a child of each branch the parent of the next branch. These sequences are summarized into a single node labeled CASCADE along with the depth (number of branches) and width (average branching factor) of the summarized branches. See Figure 2 for an example of the features summarized by these three phases. The final phase, *clean*, targets the ubiquitous empty nodes (i.e., nodes for which we lack associated data) shown in black in Figure 2. We eliminate them by merging them with their parent node, doing so repeatedly until they all have been merged into nodes that contain data. In addition to making the trees more concise, this step allows us to reason more fully over the trees since all nodes are guaranteed to contain data. This second stage of summarization further reduced the number of nodes in the median tree by another order of magnitude to about 300 nodes. Summarization similarly reduces the space required to store the data by two orders of magnitude.

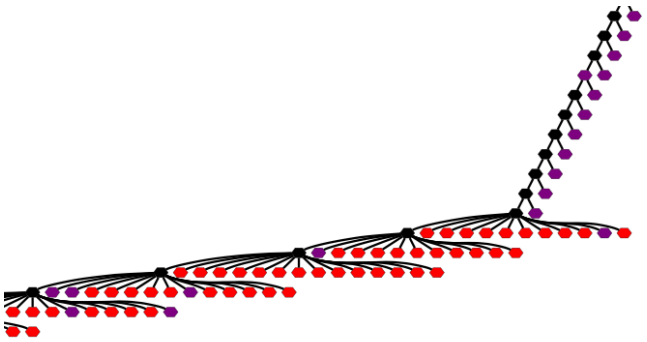


Figure 2: A solution tree after only the first stage of summarization. The non-black node color represents the score of the solution at that node (red is worse). The black nodes are empty in that we do not have solution data corresponding to that node. This figure also shows examples of the features targeted by the second summarization stage: *prune* and *collapse* eliminate long chains like the one on the right, and *condense* combines sequences of branches like those going down to left in single CASCADE nodes.

Child-parent relationships are not the only part of the data we visually encoded in the solution trees. Nodes are colored on a continuous gradient from red to blue according to the score of the solution represented by that node (red is low-scoring, blue is high-scoring). The best-scoring node is highlighted as a yellow star. Edges are colored on a continuous gradient from light to dark green according to the time the corresponding transition took place, and the children of each node are arranged left to right in chronological order. Finally, use of automation via recipes is an important aspect of problem-solving in *Foldit*. Since the logged solution snapshots contain a record of which recipes have been used at that point, we can use this to annotate nodes where a recipe was triggered. The annotations consist of the id of that recipe (a 4 to 6 digit number) and the number of times it was started.

One major weakness in the data available to us is the lack of a consistent way to determine when the execution of a recipe ended (some recipes save and restore, possibly being responsible for multiple nodes in the graph beyond where they were triggered). We partially address this by further annotating a node with the label MANUAL whenever the solver took a manual-only action at that node. This indicates that no previously triggered recipe continued past that node because no recipe could have performed the manual-only action. Since nodes in the summarized trees can represent many individual steps, it is possible for them to have several of these recipe and manual action annotations.

5. RESULTS

Using visualized solution trees for a large set of solvers across our sample of 11 puzzles, we identify a set of six prominent patterns in solvers' problem-solving behavior. These patterns do not encompass all solving behavior in *Foldit*, but instead capture key instances of strategic behavior in three categories: exploration, optimization, and human-computer collaboration. Future work is needed to generate a comprehensive survey of the strategic patterns in these and other categories. In this analysis, our focus is on identifying a small, diverse set of commonly occurring patterns to both provide initial

insight into problem-solving behavior, and to demonstrate the potential of our approach. In addition to identification, we also perform a quantitative comparison of how these patterns are employed by high-performing and lower-performing solvers to gain an understanding of how these patterns contribute to success in an open-end environment like *Foldit*.

5.1 Problem-Solving Patterns

Exploration. *Foldit* solvers are confronted with a highly discontinuous solution space with many local optima, creating a trade-off between narrowly focusing their efforts or taking the time to explore a broader range of possibilities. In our first two patterns, we examine the broader exploration side of this trade-off at two different scales. Taking the macro-scale first, we identify a pattern where solvers make significant progress on distinct branches of the tree (see Figure 3 for an example). We interpret this pattern as the solver investigating multiple hypotheses about the puzzle solution, using multiple instances of the game client or *Foldit*'s save and restore features to deeply explore them all. We call this the *multiple hypotheses* pattern.

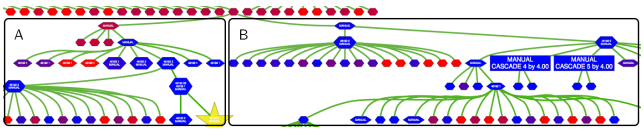


Figure 3: An example of the *multiple hypotheses* pattern. The two hypotheses branch out one of the nodes at the top and continue to the left (A) and right (B).

At the micro-scale, solvers very frequently generate a large number of possible next steps (i.e., a branch with a large number of children), but most often proceed to explore only one of them further. This is natural given the iterative refinement needed to successfully participate in *Foldit*. Hence, solvers that exhibit a pattern of much more frequently exploring multiple local possibilities demonstrate an unusual effort to explore more broadly. We call this the *inquisitive* pattern. Figure 4 shows an example of this behavior.

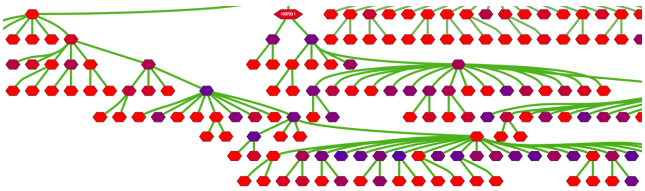


Figure 4: An example of the *inquisitive* pattern. Note how frequently multiple children of the same node are explored when compared to the tree in Figure 3.

Optimization. Navigating the extremely heterogeneous solution space is the primary challenge in *Foldit*, so we look closely at how solvers attempt to optimize their solutions, digging deeper into solvers' approach to exploration than the previous two patterns. We identify two related patterns describing solvers' fine-grained approach to optimization. The solution spaces of *Foldit* puzzles contain numerous local optima that solvers must escape, and we identify an *optima escape* pattern highly suggestive of a deliberate attempt to escape a local optima. This pattern occurs when a solver has a high-scoring node with a low-scoring child, and then chooses to explore from the low-scoring child. The solver was willing to ignore the short-term drop in score to try and reach a more beneficial state in the long-term. Figure 5 gives an example of this pattern.

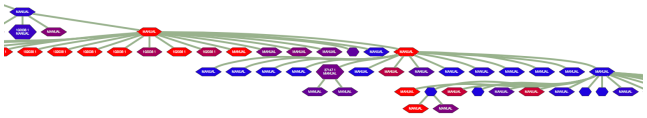


Figure 5: An example of the *optima escape* pattern. The solver transitions from a relatively high-scoring (i.e., blue) state in the upper left to a low-scoring (i.e., red) state. What makes this an example of the pattern is that exploration from the low-scoring state. In this case, the perseverance paid off as the solver reaches even higher-scoring states in the lower right.

In the other direction, we identify the *greedy* pattern in which solvers exclusively explore from the best-scoring of the available options. Obviously, some amount of greedy exploration is necessary in order to refine solutions, but in its extreme form deserves recognition as a pattern with significant potential impact on problem-solving success. Naturally, these two patterns do not cover all the ways solvers explore the problem space, but they do characterize specific strategic behavior of interest in this analysis.

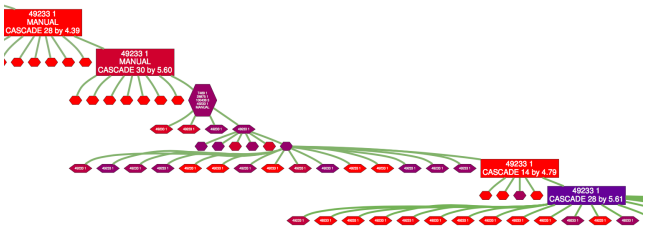


Figure 6: An example of the *repeated recipe* pattern. At three points in this solution tree snippet, the solver applies recipe 49233 to every child of a node.

Human-computer collaboration. Human-computer collaboration is a vital part of *Foldit*, and managing the trade-off between automation and manual intervention is a key feature of solving *Foldit* puzzles. We identify two patterns that each focus on one side of this trade-off. The first, the *manual* pattern, corresponds to extended sections of exclusively manual exploration. Since recipe use is very common, extended manual exploration represents a significant investment in the manual intervention side of the trade-off. Limitations with *Foldit* logging data prevent us from capturing all the manual exploration (i.e., it is not always possible to determine whether an action was performed by a solver manually or triggered as part of an automated recipe), but what can be captured is still an important dimension of variance among problem-solving behavior.

Our final pattern concerns recipe use. Some solvers apply a recipe to every child of a node periodically throughout their solution tree, using it as a clean-up or refinement step before continuing on (see Figure 6). We call this the *repeated recipe* pattern. Recipe use is very diverse and frequently doesn't display any specific structure, making this pattern interesting for its regimented way of managing some of the automation while solving.

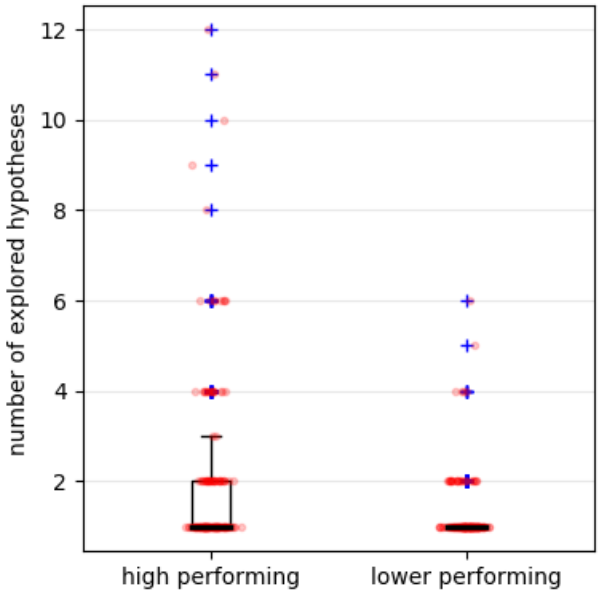


Figure 7: The number of hypotheses pursued in each solution tree for high- and lower-performing solvers. High-performing solvers frequently pursue two or more hypotheses, whereas lower-performing solvers most often pursue just one. Red circles show the distribution of individual solvers.

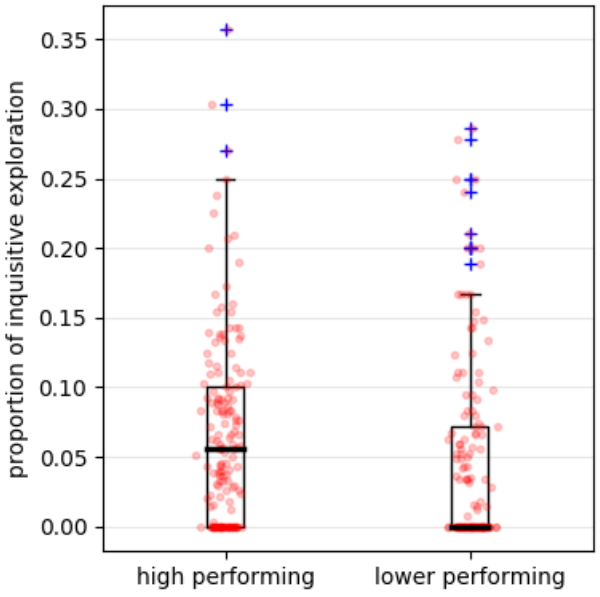


Figure 8: The proportion of all the branches in a solver's solution tree in which the solver explored more than one child for high- and lower-performing solvers. Red circles show the distribution of individual solvers.

5.2 Problem-Solving Patterns and Solver Performance

To understand how the patterns we identify relate to skillful problem-solving in an open-ended domain like *Foldit*, we compare their use among high-performing solvers to that among lower-performing solvers. Specifically, we analyze the occurrence of these patterns in the 15 best-scoring solutions from each puzzle and compare that to the occurrence in solutions from each puzzle ranked from 36th to 50th. Though it varies somewhat between puzzles, in general the solutions ranked 36th to 50th represent a *middle ground* in terms of quality. They fall outside the puzzle's state-of-the-art solutions, but remain well above the least successful efforts. Throughout these comparisons we use non-parametric Mann-Whitney U tests with $\alpha = 0.008$ confidence (Bonferroni correction for six comparisons, $\alpha = 0.05/6$), as our data is not normally distributed. For each test, we report the test statistic U, the two-tailed significance p, and the rank-biserial correlation measure of effect size r. In addition, since some of the metrics we compute may not apply to all solution trees (e.g., the tree contains no branches where the inquisitive pattern can be evaluated), we report the number of solvers involved in the comparison n for each test (the full sample is $n = 330$).

We find high-performing solvers explore more broadly than lower-performing solvers. For the *multiple hypotheses* pattern, high-performing solvers pursued significantly more hypotheses than lower-performing solvers ($U = 10569$, $p = 0.000014$, $r = 0.217$, $n = 330$) (see Figure 7). For the *inquisitive* pattern, we compute the proportion of each solver's exploration that matches the pattern (i.e., of all the branches in a solver's solution tree, in what fraction of them did the solver explore more than one child) and find high-performing solvers explore inquisitively more often than lower-performing solvers ($U = 9343$, $p = 0.000295$, $r = 0.231$, $n = 313$) (see Figure 8).

We also find high-performing solvers work harder to avoid local optima. For the *optima escape* pattern, we compute the number of times this behavior occurs in each solution and find that high-performing solvers engage in this behavior more than lower-performing solvers ($U = 11183.5$, $p = 0.00185$, $r = 0.173$, $n = 330$) (see Figure 9). For the *greedy* pattern, we compute the proportion of each solver's exploration that matches the pattern (i.e., of all the branches in a solver's solution tree, in what fraction of them did the solver only explore the best-scoring child). While high-performing solvers engaged in greedy optimization less often than lower-performing solvers, the difference was not significant ($U = 9079$, $p = 0.0158$, $r = -0.163$, $n = 295$) (see Figure 10).

Finally, we find no significant difference between high- and lower-performing solvers in the frequency they manually explore and employ recipes. For the *manual* pattern, we compute the number of manual exploration sections in each solution and find no significant difference between high- and lower-performing solvers ($U = 13334$, $p = 0.789$, $r = 0.014$, $n = 330$). For the *repeated recipe* pattern, we computed the median frequency of recipe use along all paths in the solution (i.e., for each path from the root to a leaf, in what fraction of the nodes did the solver trigger at least one recipe) and though lower-performing solvers used recipes more frequently, the difference between high- and lower-performing solvers was not significant ($U = 11342$, $p = 0.0140$, $r = -0.157$, $n = 329$).

6. DISCUSSION

The results from our analysis of our solution tree visualizations illuminate some key problem-solving patterns exhibited by individual *Foldit* solvers. Namely, how broadly an individual explores, both on a macro- and micro-scale, how actively an individual avoids

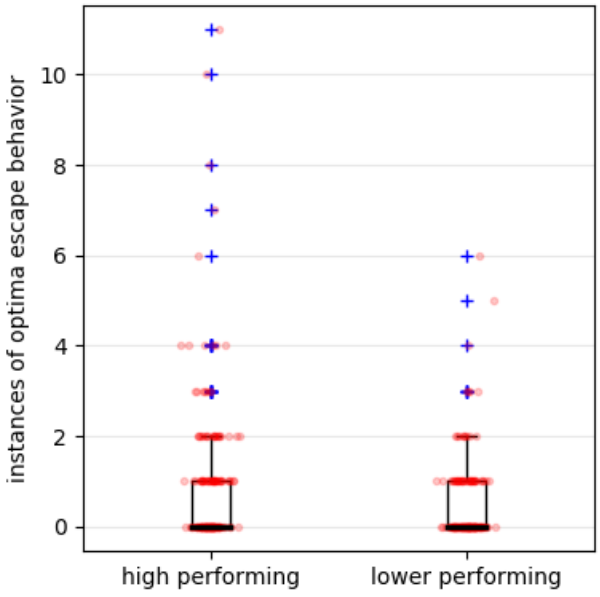


Figure 9: The number of times in each solution a solver engages in *optima escape* behavior for high- and lower-performing solvers. Red circles show the distribution of individual solvers.

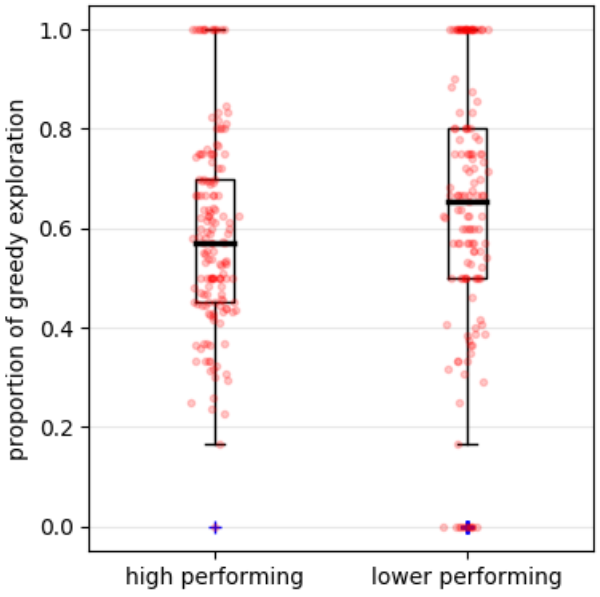


Figure 10: The proportion of all the branches in a solver's solution tree in which the solver explored only the best-scoring child for high- and lower-performing solvers. The fact that the median for both categories of solver is above 0.5 indicates that this pattern in an important part of refining solutions in *Foldit*. Red circles show the distribution of individual solvers.

local optima by engaging in less greedy optimization and actively pursuing locally suboptimal lines of inquiry, and how an individual manages the interplay between automation and manual intervention.

Comparing high- and lower-performing solvers in their application of these patterns suggests that skillful problem-solving in an open-end domain like *Foldit* involves broader exploration and more conscious avoidance of local minima. This finding that a key feature of high-skill solving behaviors is not being enamored by the current best solution and possessing strategies for avoiding myopic thinking had implications for the strategies that should be taught to develop successful problem solvers. Further work is required on other large open-ended domains to confirm this trend.

The finding that solvers of different skill use greedy exploration, manual exploration, and automation in similar amounts suggests skillful deployment of non-greedy exploration, automation, and manual intervention takes place at a more fine-grained level than overall quantity. Though this work focuses on the presence or absence of specific solving behavior, the timing and sequencing of strategic moves are likely to be critical to success. Further work is needed to investigate what differentiates effective and ineffective use of specific solving strategies.

The *Foldit* dataset itself presented significant challenges for our analysis, and we addressed these through an iterative visualization-based methodology. This process served as a design method for generating a visual grammar to describe a complex problem-solving process. We do not study the generalization of this approach to other datasets and domains in this work, but the prerequisites for its application to other open-ended problem-solving domains can be concisely enumerated: (1) the logs of solver activity establish clear temporal relationships between solution states such that those states can be visualized as a progression through the solution space, (2) the solution state or associated metadata is amenable to visual encoding, so that the visualized progressions can represent fine-grained details of the solving process, and (3) deep problem-solving domain expertise is available to provide the necessary context for interpreting and summarizing the visualized structures.

Our chosen subset of *Foldit* data represents only a small fraction of the total available data. In particular, we limited our analysis to a sample of similar prediction puzzles, and compared specific ranges of high- and lower-performing solvers. Though these choices are well-motivated, it is an important question for future work as to whether our results hold across different datasets and groups of comparison. More broadly, *Foldit* supports numerous variations on the prediction and design puzzle archetypes, which offers an exciting opportunity to study problem solving across a number of related contexts with varying goals, constraints, inputs, and tools.

7. CONCLUSION

Gaining a better understanding of key patterns in problem-solving behavior in complex, open-ended environments is important for deploying this kind of activity in an educational setting at scale. In this work, we identified six key patterns in problem-solving behavior among solvers of *Foldit*. The protein folding challenges in *Foldit* present rich, completely open, heterogeneous solution spaces, making them a compelling domain in which to analyze these patterns. To facilitate the identification of these patterns, we used an iterative methodology to design visualizations of solvers' problem-solving activity as solution trees. The size and complexity of the *Foldit* data required us to develop domain-specific techniques to summarize the solution trees and render them tractable for analysis while preserving the salient problem-solving behaviors. Finally, we compared the

occurrence of the patterns we identified between high- and lower-performing solvers. We found that high-performing solvers explore more broadly and more aggressively avoid local optima. We also found that both categories of solvers employ automation and manual intervention in similar quantities, inviting future work to study how these tools are used at a more fine-grained level.

We have only scratched the surface in our analysis of a subset of *Foldit* data. Two integral aspects of the *Foldit* environment are not within the scope of this work: collaboration and expert feedback. We only considered solutions produced by individual solvers, but *Foldit* solver can also take solutions produced by others and try and improve them. This collaborative framework may involve specialization and unique solving strategies, and deserves careful study. Expert feedback comes into play for design puzzles, where biochemists will select a small number of the solutions to try and synthesize in the lab. Experts will also impose additional constraints on future design puzzles to try and guide solutions toward more promising designs. The interaction of these channels for expert feedback and problem-solving behavior is an important topic for future research. Also outside the scope of this work is how individual solvers change their problem-solving behavior over time. Many solvers have been participating in the *Foldit* community for many years, and studying how their behavior evolves could yield insights into the acquisition of high-level problem-solving skills.

Looking more broadly at the impact of this work, our methodology and analysis can serve as a first step toward discovering the scaffolding necessary to develop high-level problem-solving skills. These results could contribute to a hint generation system, where solvers could be guided toward known effective strategies, or a meta-planner component in *Foldit* that could tailor the parameters of particular puzzles to optimize the quality of the scientific results. In all of these cases, this work contributes to the necessary foundational understanding of the problem-solving behavior involved.

8. ACKNOWLEDGEMENTS

This work was supported by the National Institutes of Health grant 1UH2CA203780, RosettaCommons, and Amazon. This material is based upon work supported by the National Science Foundation under Grant No. 1629879.

9. REFERENCES

[1] E. Andersen, Y.-E. Liu, E. Apter, F. Boucher-Genesse, and Z. Popović. Gameplay analysis through state projection. In *Proceedings of the fifth international conference on the foundations of digital games*, pages 1–8. ACM, 2010.

[2] S. Cooper, F. Khatib, I. Makedon, H. Lu, J. Barbero, D. Baker, J. Fogarty, Z. Popović, et al. Analysis of social gameplay macros in the foldit cookbook. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 9–14. ACM, 2011.

[3] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[4] M. Eagle, D. Hicks, B. Peddycord III, and T. Barnes. Exploring networks of problem-solving interactions. In *Proceedings of the 5th Conference on Learning Analytics And Knowledge*. ACM, 2015.

[5] C. B. Eiben, J. B. Siegel, J. B. Bale, S. Cooper, F. Khatib, B. W. Shen, B. L. Stoddard, Z. Popovic, and D. Baker. Increased diels-alderase activity through backbone remodeling guided by foldit players. *Nature biotechnology*, 30(2):190–192, 2012.

[6] M. H. Falakmasir, J. P. Gonzalez-Brenes, G. J. Gordon, and K. E. DiCerbo. A data-driven approach for inferring student proficiency from game activity logs. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 341–349. ACM, 2016.

[7] C. Geigle, C. Zhai, and D. C. Ferguson. An exploration of automated grading of complex assignments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 351–360. ACM, 2016.

[8] M. L. Gick. Problem-solving strategies. *Educational psychologist*, 21(1-2):99–120, 1986.

[9] J. G. Greeno. Natures of problem-solving abilities. *Handbook of learning and cognitive processes*, 5:239–270, 1978.

[10] E. Harpstead, C. J. MacLellan, K. R. Koedinger, V. Aleven, S. P. Dow, and B. Myers. Investigating the solution space of an open-ended educational game using conceptual feature extraction. In *Proceedings of The 6th Conference on Educational Data Mining*, 2013.

[11] W. Hung, D. H. Jonassen, R. Liu, et al. Problem-based learning. *Handbook of research on educational communications and technology*, 3:485–506, 2008.

[12] M. Johnson, M. Eagle, and T. Barnes. Invis: An interactive visualization tool for exploring interaction networks. In *Proceedings of the 6th Conference on Educational Data Mining*, 2013.

[13] D. H. Jonassen. Toward a design theory of problem solving. *Educational Technology Research and Development*, 48(4):63–85, dec 2000.

[14] D. A. Joyner. Expert evaluation of 300 projects per day. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 121–124. ACM, 2016.

[15] F. Khatib, S. Cooper, M. D. Tyka, K. Xu, I. Makedon, Z. Popović, and D. Baker. Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences*, 108(47):18949–18953, 2011.

[16] F. Khatib, F. DiMaio, S. Cooper, M. Kazmierczyk, M. Gilski, S. Krzywda, H. Zabranska, I. Pichova, J. Thompson, Z. Popović, et al. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature structural & molecular biology*, 18(10):1175–1177, 2011.

[17] L. Malkiewich, R. S. Baker, V. Shute, S. Kai, and L. Paquette. Classifying behavior to elucidate elegant problem solving in an educational game. In *Proceedings of the 9th Conference on Educational Data Mining*, 2016.

[18] E. Rowe, R. S. Baker, and J. Asbell-Clarke. Strategic game moves mediate implicit science learning. In *Proceedings of the 8th Conference on Educational Data Mining*, 2015.

[19] H. A. Simon. Information-processing theory of human problem solving. *Handbook of learning and cognitive processes*, 5:271–295, 1978.

[20] K. Tóth, H. Rölke, S. Greiff, and S. Wüstenberg. Discovering students' complex problem solving strategies in educational assessment. In *Proceedings of the 7th Conference on Educational Data Mining*, 2014.

[21] G. Wallner and S. Kriglstein. Visualization-based analysis of gameplay data–a review of literature. *Entertainment Computing*, 4(3):143–155, 2013.

The Antecedents of and Associations with Elective Replay in an Educational Game: Is Replay Worth It?

Zhongxiu Liu
North Carolina State University
zliu24@ncsu.edu

Christa Cody
North Carolina State University
cncody@ncsu.edu

Tiffany Barnes
North Carolina State University
tmbarnes@ncsu.edu

Collin Lynch
North Carolina State University
cflynch@ncsu.edu

Teomara Rutherford
North Carolina State University
taruther@ncsu.edu

ABSTRACT

Replayability has long been touted as a benefit of educational games. However, little research has measured its impact on learning, or investigated when students choose to replay prior content. In this study, we analyzed data on a sample of 4,827 3rd-5th graders from ST Math, a game-based educational platform integrated into classroom instruction in over 3,000 classrooms across the U.S. We identified features that describe elective replays relative to prior gameplay performance, and associated elective replays with in-game accuracy, confidence, and general math ability assessments outside of the games. We found some elective replay patterns were associated with learning, whereas others indicated that students were struggling in the current educational content. We suggest, therefore, that educational games should use elective replay behaviors to target interventions according to when and whether replay is helpful for learning.

Keywords

Educational Games, Serious Game Analytics, Replayability

1. INTRODUCTION

"Replayability is an important component of successful games." [15] In most games, there are two types of plays: play and replay to pass a level (*pass attempts*) and replay after passing a level (*elective replay*). In this paper, we investigate the latter. Elective replay (*ER*) is particularly interesting because the motivations behind a student's decision to replay and the impact of those replays are relatively unknown. This paper explores potential associations between elective replay and student characteristics and performance in the domain of educational games.

Replayability has been touted as a benefit of educational games [9]. Replayability encourages players to engage in repeated judgement-behavior-feedback loops, where users make decisions based on the situation and/or feedback, act on those decisions, and receive feedback based on their actions [18]. In the RETAIN model designed by Gunter et al. [10] to evaluate educational games, replayability is a criteria for naturalization – an important component in helping students make their knowledge automatic, reducing the cognitive load of low-level details to allow for higher order thinking. In the RETAIN model, "replay is encouraged to assist in retention and to remediate shortcomings." [10] Meaningful elective replay is often encouraged by game features such as score leaderboards, which inspire students to replay for higher scores [4]. Because higher scores typically require a deeper understanding of the educational content in a well-designed game, encouraging elective replay may promote mastery. Games with replay also allow the student to be exposed to more material and give them more freedom to control their learning. Studies have shown that giving students control over their learning process can increase motivation, engagement, and performance [6, 8].

However, few studies have investigated when students choose to replay, why they do so, or have measured the outcomes associated with elective replay. One reason is that educational game studies are often comparatively brief, so replayability is often minimally assessed with post-game questionnaires asking about students' intention for future play [14, 5]. Consequently, there is a need to investigate elective replay with actual logged actions in a game setting where students have sufficient time and freedom to replay.

This work analyzed gameplay logs from a series of math games within the year-long supplemental digital mathematics curriculum Spatial Temporal (ST) Math. We analyzed gameplay data from 4,827 3rd-5th graders throughout the 2012-2013 school year. Our data contained 37,452 logged elective replays, accounting for 1.48% of the logged play. We analyzed gameplay and elective replay features in association with students' demographic information, in-game math objective tests, and the state standardized math test. We sought to answer three research questions: Q1: What are the characteristics of students who engage in elective replay, Q2: What gets replayed, and under what circumstances? And Q3: Is elective replay associated with improvements in students' accuracy on math objectives, confidence, and

general math ability?

2. RELATED WORK

2.1 Factors Influencing Elective Replay

Few empirical studies have investigated the motivations behind elective replay in educational games. Burger et al. [5] studied the effect of verbal feedback from a virtual agent on replay in the context of a brain-training game. They found that elaborated feedback increases, whereas comparative feedback decreases, the students' interest in future replay. They also found that negative feedback generated an immediate interest in replay, whereas positive feedback created long term interest in the educational content. In another study, Plass et al. [14] compared three conditions in a math game: working individually, competing with another player, or collaborating with a peer. The study showed that both competition and collaboration modes heightened students' intention to replay when compared with the individual mode, with the latter result being statistically significant. However, both studies measured replay via questionnaires asking the students' desire to play the entire game again instead of observed replay behavior. Moreover, these studies sought to understand replay only from the angle of game design, and did not address the connections, if any, between student characteristics and interest in replay.

Other studies suggest elective replay is a habitual behavior that arises from individual need, although these studies did not directly investigate replay. Bartle [3] found one type of player who is primarily motivated by concrete measurements of success. In ST Math, these *achiever-type* players may largely use replay to get better 'scores' (losing fewer lives when passing a level). Mostow et al. [12] observed a student in a reading tutor who used the learner-control features to spend the majority of time replaying stories or writing "junk" stories instead of progressing to new material. Thus, some students may also use replay as a form of work avoidance – playing already passed levels instead of solving the current problem or moving on. Sabourin et al. [17] found that students in an educational game used off-task behaviors to cope with frustration, implying that off-task behavior can be a productive self-regulation of negative emotions. In ST Math, when students get frustrated with the current educational content but still have to play the game in the classroom, they may replay already learned content as a mental break from the current task. These studies showed that the circumstances of replay and students' characteristics influence their decisions to replay and its outcomes.

2.2 The Outcomes of Replay

Despite the believed benefits of replayability [9, 18, 10, 4], few studies have investigated the educational impact of elective replay. Boyce et al. [4] evaluated the effects of game elements that were designed to motivate gameplay and elective replay. These included a leaderboard that shows each student's rank based upon their score, a tool for creating custom puzzles, and a social system for messaging among players. The experimental design required students to play the game in one session, and to replay the game as more features were added in the subsequent sessions. The study found a sharp increase in test scores as these features were added to the game. The authors concluded that features designed to increase replayability can increase learning gains. However, this result may be due to increased time on task as the same group replaying the base game with new features. In another study, Clark et al. [7] analyzed logged student-initiated elective replay in a digital game. They found that frequency of elective replay did not correlate with learning gains, prior gaming habits/experience, or how much students liked the game. They also found that, while there was no statistically significant difference between the male and female students, males replayed more than the females. This may have been responsible for their slightly higher, although not statistically significant, "best level scores" – the highest score received on each level. These studies showed that elective replay may lead to increased learning or higher in-game performance. However, more research is needed to understand the potential educational impact of replay in educational games, particularly elective replays initiated solely by the players.

3. GAME, DATA AND FEATURES

3.1 ST Math Game

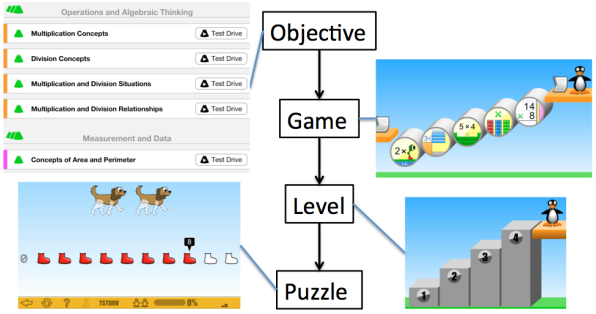


Figure 1: ST Math Content and Examples

ST Math is designed to act as a supplemental program to a school's existing mathematics curriculum. ST Math is mostly played during classroom sessions, but students have the option to play it at home. In ST Math [16], mathematics concepts are taught through spatial puzzles within various game-like arenas. ST Math games are structured at the top level by objectives, which are broad learning topics. Within each objective, individual games teach more targeted concepts through presentation of puzzles, which are grouped into levels for students to play. Students start by completing a series of training games on the use of the ST Math platform and features. They are then guided to complete the first available objective in their grade-level curriculum, such as "Multiplication Concepts." Students can only see this objective and must complete a pre-test before beginning the content. Games represent scenarios for problem-solving using a particular mathematical concept, such as "finding the right number of boots for X animals of Y legs." Each game contains between one and ten levels, which follow the same general structure of the game, but increase in difficulty. Figure 1 illustrates the hierarchy of ST Math content and examples.

As with many games, the student is given a set number of 'lives' at the start of each level. Every time they fail to

complete a puzzle correctly they lose one life. If all of their lives for a given level are exhausted, they will fail the level and be required to restart the level with a new set of lives. Once a student has passed a level, they can elect to replay it at any time. After a student has passed every level in an objective, they can take the objective post-test. Students cannot progress to the next objective until they have completed the last objective post-test. Both the objective pre- and post-tests consist of 5-10 multiple choice questions related to the objective. The post-tests parallel the pre-tests in both the question format and difficulty of the content. While answering each question in both tests, students indicate their relative confidence in their answer (low/high).

3.2 Data

MIND Research Institute (MIND), the developers of ST-Math, collected and provided to the researchers gameplay data from 4,827 3rd-5th graders during the school year 2012-2013. These students came from 17 schools and 221 classrooms. Table 1 summarizes students' demographic information. These demographic data, together with students' state standardized test scores in 2012 and 2013, were matched to gameplay data through anonymized IDs.

Table 1: Populations' Demographics Information

	Grade3	Grade4	Grade5
#Students	1567	1528	1732
Male	50.6% na:2.9%	50.1% na:2.0%	52.2% na:3.5%
Eligible for Reduced Lunch	80.7% na:2.9%	77.8% na:2.1%	81.4% na:3.2%
Hispanic or Latino	84.7% na:2.8%	82.3% na:1.9%	83.5% na:3.1%
English Language Learner	66.2% na:2.9%	56.1% na:2.1%	53.0% na:3.2%
with Listed Disability	10.9% na:2.1%	11.5% na:1.7%	11.9% na:2.8%

This gameplay data includes pre- and post-tests for each objective and the number of level attempts. For each pre- and post-test, ST Math logged students' accuracy and self-reported confidence level (1 for 'high' and 0 for 'low') for each question. For each play at a level, ST Math logged the student's ID, timestamp, and the number of puzzles completed. From these data, we identified ER as plays made after a student initially passed the level. We found ERs in 89.6% of all objectives in ST Math, accounting for 1.48% of all level attempts. Among 4,827 students, 59.85% ERed at least one level, with an average of 7.84 levels (SD=12.99, 95% CI [7.37, 8.32]) across 3.06 average objectives replayed per student. In the next section, we describe the features we created to analyze ER.

3.3 Features

We created features at three different levels of granularity (from finest to largest): level, objective, and student. For the level granularity, we treated each unique student-level combination as an observation. We calculated the features by averaging all gameplay for a specific student at a specific level. For objective granularity, each unique student-objective combination was treated as a single observation. Features were created by averaging across all levels played by a specific student within a single objective. The objective granularity also included the objective pre- and post-test accuracy and confidence. For the student granularity, we treated each student as a single observation. We calculated the features by averaging across all objectives played by a student over the entire year. The student granularity also included student demographic data and state standardized math test scores. These granularities ensured that our analysis did not favor units with the majority of data logs. Each student was considered equally in our analysis, regardless of how many objectives they played. Our data contained 4,827 students and 2,524,681 plays, which yielded 1,462,660 student-level observations, and 74,985 student-objective observations.

Table 2 shows five example plays of "Division-Level3," including four pass attempts and one ER of this level, interspersed with ERs from other levels. We consider consecutive ERs as an ER Session, as these ERs are circumstanced on the same pass attempts.

Table 2: Example of ER and Pass Attempts

Play	Objective-Level	Passed?	Play Type
1	Division- Level3	No	Pass Attempt
2	Division- Level3	No	Pass Attempt
3	Division-Level1	Yes	ER (ER Session1)
4	Division- Level3	No	Pass Attempt
5	Division-Level1	Yes	ER (ER Session2)
6	Division- Level3	Yes	Pass Attempt
7	Division- Level3	Yes	ER (ER Session3)
8	Subtraction-Level1	No	ER (ER Session3)

3.3.1 *Pass Attempt Features*

We defined performance to be the percentage of puzzles a student completed before losing all lives on the level. Pass attempts are plays prior to ER, where we assumed students play with the intention of passing the level. Pass attempt features included: performance when a student first attempted a level (*1st pass attempt performance*), number of attempts taken to pass a level (*# pass attempts*), and average performance of all pass attempts (*average pass attempt performance*). At the student granularity, students took an average of 1.91 (sd=0.89) attempts to pass each level, with average performance of 0.80 (sd=0.10) on the first pass attempt, and 0.87 (sd=0.07) on all pass attempts (indicating overall improved performance on later attempts).

3.3.2 *Elective Replay Features*

Table 3 shows ER features that describe ER from three angles: (I) the frequencies of ER, (II) the performance of ER, and (III) the circumstances of ER in terms of the ER's prior plays. To summarize, the majority of ERs had higher performance than their levels' first attempt, and resulted in another pass of their levels. Levels that were ERed had similar performance compared to levels that weren't ERed, but levels that were followed(54.65%) or interrupted (54.35%) by ER had much lower performance than those that weren't followed or interrupted by ER. Most ERs' immediately prior pass attempts were from different levels or objectives. There were few instances (9.80%) where students passed a level and immediately ERed it following the pass.

Table 3: Elective replay (ER) Features and their Descriptive Statistics among Students who Electively Replayed, Collapsed to the Student Granularity.

ER Features	Descriptive Stats
I. Frequencies of ER	
% ER out of all plays	M=2.40%, SD=4.26%
% Objectives that have been electively replayed	M=22.94%, SD=20.89%
% Objectives whose pass attempts were interrupted/followed by ER	M=19.48%, SD=17.57%
II. Performance of ER	
Performance of ER	M=0.71, SD=0.28
% ERs performed better than the level's first attempt	M=71.96%, SD=31.44%
% ERs that result in another pass of the level	M=60.36%, SD=35.51%
III. Circumstances of ER	
The Replayed Level E.g. "Division-lvl1," "Division-lvl3," and "Subtraction-lvl1" in Table 2	
Pass Attempts Features	M=0.79, 1.98, 0.87 for 1st performance, #pass attempts, and avg performance
The Immediately-Prior play of the ER E.g. Play 2 is the immediately-prior play of play 3 in Table2	
Performance on the immediately-prior play	M=0.63, SD=0.29
% ERs whose immediately-prior plays is also an ER	M=0.31, SD=0.28
% ER whose immediately prior pass attempt is on the same level	M=9.80%, SD=23.84%
% ..... on a different level in the same objective	M=40.75%, SD=39.09%
% ..... on a different objective	M=49.44%, SD=40.76%
The Immediate Prior Pass Attempts followed or interrupted by ER and ER Session E.g. "Division-lvl3" for all ER Sessions in Table 2	
Pass Attempts Features	M=0.51, 3.62, 0.55 for 1st performance, #pass attempts, and avg performance
% ER sessions whose prior pass attempt passed the level	M=45.65%, SD=40.69%

Note. statistics are reported at the student granularity, which are calculated through averaging across all objectives played by a student, and then averaged across all students who electively replayed. This means each student contributes equally to the average, regardless of how many objectives s/he played.

3.3.3 Student Grouping From ER Features

We created student groups to encapsulate the circumstances under which ER occurred, based on students' majority ER and ER sessions. Based on prior literature, we hypothesized that ER is a habitual behavior that arises from individual needs, such as gaining higher scores [3], avoiding progress on the current task [12], or taking a mental break from negative emotions [17]. Thus, grouping students based upon the circumstances of replay based on their majority behaviors provides high level profiles to investigate characteristics of students who engaged in ER and benefited from ER.

We characterized ER by the timing relative to the student's current learning objectives and gameplay. The first grouping describes whether the majority ER sessions started before (Group B) or after (Group A) passing the previous attempted level (current learning objective). If there is a tie between the two types of replay session, the student belongs to neither group. For example, Table 2 describes a group B student, who has two replay sessions before passing "Division-level3," and one replay session after passing this level but before moving on to the next level.

The second grouping describes whether an ER followed plays on the same level (SL), a different level under the same objective (DLSO), or a different objective (DO). For our example in Table 2, the student's pass attempts on "Division-Level3" was interrupted twice on the third and fifth plays, by replays on "Division-level1"(DLSO). After passing "Division-level3", the student replayed the same level(SL) once during the seventh play, and a different objective " Subtraction-level1" (DO) once during the eighth play. This Group B student had two DLSO replays, one SL, and one DO replays. Thus, this student also belongs to Group DLSO, because the two groupings are independent of each other.

4. METHODS & RESULTS

4.1 Who Engaged in Elective Replay?

We first investigated the demographic characteristics of students who engaged in elective replay. We found that males did so more often than females (male: 63.2%, female: 57.0%, $c2(1, N=4827) = 17.99$, $p<.001$). We also found that English Language Learners (ELL) did so more often than their non-ELL peers (ELL: 62.3%, non-ELL: 57.1%, $c2(1, N=4827) = 12.69$, $p<.001$), as did students with reported disabilities (disability: 68.7%, non disability: 59.1%, $c2(1, N = 4827) = 18.17$, $p<.001$). There were no statistically significant differences in the frequencies of ER based on race when operationalized as Hispanic/non Hispanic, or based on free/reduced lunch eligibility. The frequency of ER was not found to be correlated with other out-of-game student factors, such as state standardized math test scores.

The frequency of ER was also not correlated with in-game pre-test accuracy and confidence at the objective granularity. Next, we investigated the gameplay characteristics of students who electively replayed. We first separated students into groups based on their replay patterns. The first

Table 4: Mann-Whitney U Tests Comparing Gameplay Characteristics between ER Pattern Student Groups

Group (# students)	Pre-test Accuracy	Pre-test Confidence	Avg Pass Attempts' Performance	Avg 1st Attempt Performance	#Pass Attempts	ER Performance
Base:No ER (N=1938)	M=0.61 SD=0.17	M=0.75 SD=0.23	M=0.88 SD=0.08	M=0.81 SD=0.11	M=1.82 SD=0.84	NA
ER (N=2889)	*M=0.57 SD=0.17	M=0.74 SD=0.24	*M=0.87 SD=0.07	*M=0.80 SD=0.10	*M=1.92 SD=0.78	M=0.72 SD=0.29
Group A (N=1114)	M=0.62 SD=0.16	M=0.77 SD=0.22	*M=0.90 SD=0.05	*M=0.84 SD=0.08	*M=1.62 SD=0.52	*M=0.77 SD=0.27
Group B (N=1464)	*M=0.52 SD=0.17	*M=0.72 SD=0.25	*M=0.84 SD=0.07	*M=0.75 SD=0.09	*M=2.28 SD=1.09	*M=0.67 SD=0.29
Group SL (N=173)	M=0.61 SD=0.17	M=0.75 SD=0.23	M=0.88 SD=0.07	M=0.81 SD=0.09	M=1.82 SD=0.81	*M=0.84 SD=0.29
Group DLSO (N=983)	*M=0.54 SD=0.18	M=0.73 SD=0.24	*M=0.84 SD=0.08	*M=0.76 SD=0.10	*M=2.27 SD=1.16	*M=0.67 SD=0.32
Group DO (N=1399)	*M=0.58 SD=0.16	M=0.75 SD=0.23	M=0.88 SD=0.06	M=0.81 SD=0.08	M=1.80 SD=0.71	M=0.73 SD=0.26

Note. 1) Green and red indicate statistically significances higher and lower than the base class, with $*p < .001$, $+p < .01$ 2) Group A, B: most ER sessions happened before (B), after (A) passing the prior non-replay level. Group SL, DLSO, DO: most ER followed pass attempts on the same level(SL), different level in same objective(DLSO), or different objective (DO)

5 columns of Table 4 shows the results of Mann-Whitney U tests with Benjamini-Hochberg correction to compare each group in-game performance to the students who never electively replayed any levels (the Base group). The last column compares the averaged ER performance of each group to the rest of students who electively replayed.

Compared to the base group, students for whom most replays happened before passing the prior non-replay level (Group B) and students for whom most replays followed a different level on the same objective (Group DLSO) started with significantly lower pre-test scores and did worse in gameplay, as measured by the three pass attempt features described in section 3.3.2. For example, students in Group B started with lower accuracy and confidence at pre-test, took an average 0.5 more attempts to pass a level, and had lower performance on the 1st pass attempt and all pass attempts (including the 1st). It seems that Group B students who replayed earlier levels before passing the current one had less prior knowledge, and struggled more in the game. By contrast, students in Group A, for whom most replay happened after passing the current level, did slightly better in gameplay compared to students who never electively replayed (the Base group). Because these students started with pre-test scores that were not statistically significantly different from the base group, their replay patterns are associated with higher gameplay performance.

4.2 What Gets Replayed, and When?

Next, we studied what levels get replayed, and under what circumstances. We used a decision tree classifier which allowed us to identify which factors are most important in relative to ER. Our goal was not to find precise predictive models, but to augment our understanding of performance and its relationship to ER. We used R's *rpart* package with parameters minsplit=5% and cp=0.02 to build trees to classify levels that were replayed from levels that were not replayed, and levels whose pass attempts were interrupted or followed by replay from levels that were not interrupted or followed by replay. We randomly undersampled the majority class (levels without replay, levels were not interrupted or followed by replay), so that each class represented half of the observations. We used pass attempt features at the level granularity together with pre-test results, objective, and demographic information to build our tree. We used 10-fold cross validation to access the trees' accuracies.

Table 5 reports the trees and the importance of the features. We found that a student's performance on a particular level influenced whether replay happened during/after the level's pass attempts. For example, a student was more likely to replay a different level under the same objective (DLSO) if they took more than two attempts to pass the current level. This result is related to the previous result in Table 4, showing that, at the student level, those with lower gameplay performance were more likely to replay another level under the same objective.

On the other hand, the objective to which a level belongs influences whether or not a level would be ERed. We built trees to predict if a level is replayed following the same level (same condition of the last row in Table 5, N=1,776), the same objective but a different level (N=12,616), or a different objective (N=31,852). For all three conditions, the trees only contains a single node – objective, with accuracy of 55.2%, 62.0%, and 66.9% respectively. This ER decision could have been influenced by either the content or timing of the objectives. In our tree node, we noticed that many objectives with a higher chance of ER occurred earlier in the curriculum, this could be because students had more time in which these objectives were available for ER. Our tree model also had only 55.2% accuracy when predicting whether a level would be ERed following the pass attempts of itself. One explanation is that we do not have puzzle granularity data on how many lives a student actually lost. From prior literature [4] [7], students may replay the same

Table 5: Decision Trees to Predict Levels whose Pass Attempts were Interrupted or Followed by ER

Condition: interrupted/followed by	**Trees**
ER from a different level in the same objective (N=8,094)	77.8% accuracy #pass attempts $<$ 2.5, No #pass attempts $\geq$ 2.5, Yes
ER from a different objective (N=12,506)	78.7% accuracy 1st attempt performance $\geq$ 0.94 -objective group A, No -objective group B, Yes 1st attempt performance $<$ 0.94 -objective group A —# pass attempts $<$ 6.5, No —# pass attempts $\geq$ 6.5, Yes -objective group B, Yes
ER on the same level (N=1,766)	55.2% accuracy objective group A, No objective group B, Yes

Note. Trees are presented in text format. For example, the first tree shows that if a student passed a level with less than 2.5 pass attempts, the tree predicts this student will not replay another level during/after this level.

level following it pass attempts to get a better score, which means losing fewer lives (making fewer errors) at a level. As shown in Table 4, Group SL students who performed most of their ERs after the same level also achieved the highest ER performance.

4.3 Is Elective Replay Associated with Gains?

In this section we will address our second research question. As part of our analysis we considered three gain scores: accuracy gain, confidence gain, and math gain. The first two were measured by in-game pre- and post-tests. Recall that both before and after a student attempts an objective, ST Math logs the students' correctness and confidence scores on each question on the pre- and post-tests. We averaged these scores across the pre- and post-test questions to compute the first two gain scores. These were assessed at the objective granularity. Math gain was calculated based upon the difference between the students' state standardized math test scores in years 2012 and 2013. This was assessed at the student granularity.

11.8% of the students were excluded from the math gain analysis due to missing state math test records. These excluded students performed statistically significantly worse in the game as measured by the three pass attempt features; this implies that we excluded weaker students. 8.5% of the objective observations were excluded from the accuracy and confidence gain analysis due to missing pre- or post-tests. These excluded observations were not statistically significantly different from the rest as measured by pass attempt features. The accuracy and confidence gains were significantly correlated (r=0.37, p<0.001), but these two gains were not strongly correlated with math gain scores at the student granularity (r<0.1, p<0.001). Table 6 reports the percentage of data points that gained, dropped (mainly for avoiding ceiling effect in this data), and did not gain for each

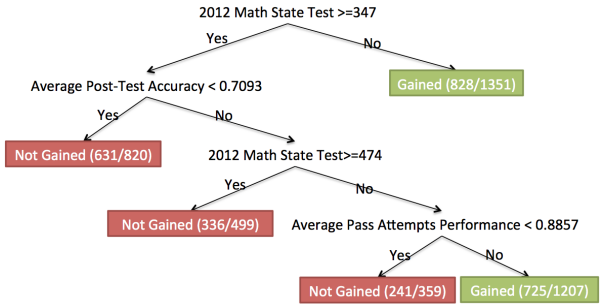


Figure 2: Decision Tree to Predict Whether a Student will Gain in State Standardized Math Test

type of gain based on the Marx and Cummings Normalization method [11].

Table 6: %Observations with Gains, No Gains, and Percentage Dropped for the Three Gains

Gain Types	**ER?**	**Gained**	**Dropped**	**No Gain**
Accuracy	ER	48.10%	8.60%	37.90%
(N=75,083)	No ER	43.70%	6.10%	36.60%
Confidence	ER	28.30%	42.60%	23.70%
(N=75,083)	No ER	26.40%	37.40%	22.70%
Math Test	ER	41.60%	0.40%	46.90%
(N=4,827)	No ER	40.80%	0.50%	45.70%

Note. 1)Observations in the 'Dropped' column (pre- and post-tests were both 0 or 1) were excluded from analysis. 2)Accuracy and Confidence Gains were measured at objective granularity, Math gain was measured at student granularity. 3)ER and no ER were collapsed across level.

We first constructed decision trees to partition our data to see which factors influence gains, using the method described in the prior section. No sampling was necessary because the groups had similar sizes. We used pass attempt features, ER features, pre-test results, and demographics. For student granularity, we also added the percentage of required objectives attempted by the student.

At the objective granularity, we found that pre-test accuracy and confidence were the only selected nodes that predicted accuracy (70.0% accuracy) and confidence gain (74.1% accuracy). Students with a pre-test accuracy of $<$ 0.71 (at least 2 questions wrong out of 5-10) had a 64.7% chance of positive accuracy gain in the same objective, while the remainder of the students had only a 25.9% chance. Students with high pre-test confidence ($\leq$0.95, indicated confidence on almost all questions) had a 62.5% chance of positive confidence gain in the same objective. It could be that these in-game tests were too easy, as 18.9% of pretests achieved full scores in accuracy and 54.5% achieved full scores in confidence.

Our decision tree for the student granularity is shown in Figure 2, with a cross-validated accuracy of 57.8%. Students who started with medium level of math abilities (2012 state test math scores $<$474, and $\geq$ 347) improved their scores when they performed well in ST Math (average pass attempts performance $>$ 0.8857). This shows that the game-

play data in ST Math has predictive power for assessment outside of the game. However, for all three gain scores, the ER features were not selected for inclusion in the decision tree nor was any correlation found with the students gains.

Table 7: Mann-Whitney U Tests Comparing Gains between ER Pattern Student Groups.

Group (# students)	Math (max=600)	Accuracy (max=1)	Confidence (max=1)
Base:No ER (N=1938)	M=31.5 SD=146.6	M=0.31 SD=0.25	M=0.33 SD=0.38
ER (N=2889)	M=27.3 SD=139.7	M=0.30 SD=0.25	M=0.32 SD=0.37
Group A (N=1114)	M=53.4 SD=167.9	*M=0.35 SD=0.24	+M=0.38 SD=0.36
Group B (N=1464)	+M=6.7 SD=109.0	*M=0.24 SD=0.25	*M=0.26 SD=0.37
Group SL (N=173)	M=46.2 SD=161.2	M=0.31 SD=0.28	M=0.31 SD=0.37
Group DLSO (N=983)	M=21.4 SD=123.0	*M=0.25 SD=0.26	*M=0.27 SD=0.37
Group DO (N=1399)	M=32.3 SD=150.6	M=0.32 SD=0.23	M=0.34 SD=0.36

Note. green and red indicate statistically significances higher and lower than the base class, with $^*p < .001$, $+p < .01$

Finally, we investigated how ER patterns relate to gains. Table 7 reports the result from separating students into 6 groups based on ER patterns and conducting Mann-Whitney U tests with Benjamini-Hochberg correction (as in the previous section). Moreover, although decision trees constructed from the complete dataset show that low pre-test results led to more gains, some ER pattern groups showed opposite trends. For example, Group B, who primarily ERed before passing the current level, started with lower pre-test scores, did worse in the game, and had less gains, which were statistically significant, in all three gain measures. The same applies to Group DLSO. These two groups of students also had the lowest ER performance.

On the other hand, the Base group and Group A (who mostly ERed after passing the current level) started with pre-test accuracy and confidence scores that are not significantly different (Table 4), but Group A did significantly better in game, and had higher gains in accuracy and confidence, which were statistically significant. Because the mean pre-test score for the Base and A groups is approximately 0.6, these students were reasonably familiar with the objective before they began playing it. The difference in accuracy and confidence gains suggest that ER after students successfully pass a level helped students learn, or implied better learning in the previous gameplay.

5. DISCUSSION AND CONCLUSIONS

This work presents a significant extension on prior studies of replay which have typically taken place over a short period of time and have assessed replay via intentional questionnaires not observed behaviors [14, 5]. This work analyzed logged student-initiated elective replay from a sample of 4,827 3rd-5th graders during school year 2012-2013 in ST Math in a natural educational setting. We sought to answer three research questions: Q1: What are the characteristics of students who electively replay? Q2: What gets replayed, and under what circumstances? And Q3: Is elective replay associated with improvements in students' accuracy on math objectives, confidence, and general math ability?

We concluded that, with over half of students who electively replayed at least one level, ER is a common behavior in ST Math. Moreover, examining elective replay can enhance our understanding about how students play and the characteristics of successful play. For example, we found that students who did poorly on the current level were more likely to electively replay a different level during/after the level's pass attempts. We also found that students who generally engaged in elective replay before passing the current level (Group B) started with lower pre-test scores, did worse during gameplay, and had the lowest objective-level accuracy and confidence gain and math gains. One explanation for this result is that weaker students used ER as a work avoidance tactic, as found in Mostow et al. [12], and that instances of ER stand in for lower motivation or engagement for the objective topic, ST Math, or mathematics overall.

On the other hand, compared to students who didn't ER, students who mostly electively replayed after passing the current level (Group A) started with pre-test scores that were not significantly different, did better in the game, and had higher learning and confidence gains. One reason could be that these students electively replayed for a better score, as we also found that students who mostly replayed the same level immediately after passing it (Group SL) had the highest ER performance. This association is especially true among achiever-type players [3] that prefer to gain concrete measurements of success. Because losing fewer lives in ST Math requires better mastery of the math content, ER may have helped these students learn. Another explanation is that these students' ERs could imply better learning during prior gameplay, as Table 4 also shows that Group A students had better pass attempt performance. Possibly, successful prior performance motivated these students to electively replay more of the game. Moreover, because successful prior performance feeds self-efficacy [2, 13], confidence gains in Group A students, who chose more ER, may be linked to electively replaying levels they have already mastered.

From the application perspective, as expected from this complex environment, our effect-sizes are too small to claim ER itself as a powerful intervention for learning. Instead, our findings suggest the potential of using ER patterns to identify weaker students and their struggling moments for intervention. For example, students with Group B ER patterns started weaker, did poorly in the game, and had lower gains in learning, confidence, and math state test scores. It may be the case that Group B ER (before passing a level) is a signal that students are struggling in current content and are in need of a mental break [17] or help. If this is the case, it would be beneficial upon detecting these ER patterns for ST Math to alert teachers or to provide interventions, such as suggesting the student to take a break or providing supplemental resources to further explain the math concepts from the pass attempts interrupted by ER. Our results also suggest avenues for experimental studies that designs a more effective ER experience, such as preventing work-avoidance

in ER. For example, changing the number of lives students have at each replay, or constraining the problems offered each time they are replayed to be isomorphic but not identical.

This work has several limitations. First, the in-game pre-post- tests may be too easy for students, as 18.9% of pretests achieved a full score in accuracy, and 54.5% achieved a full score in confidence. The high percentage of students with non-positive learning and accuracy gain could also be caused by students' slipping or guessing in multiple-choice questions (e.g., 1 incorrect answer reduces accuracy by 14%-20%). The accuracy of the pre- and post-test questions for assessing knowledge might be improved by using short answer questions. The second limitation is that we did not have puzzle granularity data on how many lives a student actually lost or the types of errors they made. Third, the grouping of students based on the majority of elective replay assumes that elective replay is a habitual and consistent behavior. Future research should investigate other groupings, as well as examining whether there were changes in how students used replay, and what caused the changes. Fourth, future work may also include creating quantified features to compare the content and game features across objectives so we may better understand how the game's content influence students' decision to engage in elective replay.

In summary, this work adds new insights to our understanding of elective replay in educational games. Our work reveals differential associations between elective replay and performance when replay is categorized by the timing in relation to the student's current learning objectives and gameplay. Our work suggests that low-performing students did not benefit from ER; high-performing students both chose ER at better times and their ERs were associated with benefits from either ER or previous gameplay, which supports the results of prior self-regulation research by Aleven et al [1]. This work presents prospects for both examining more detailed characteristics of replay and utilizing experimental manipulations.

6. ACKNOWLEDGEMENTS

This work was supported by NSF grant IUSE #1544273 "Evaluation for Actionable Change: A Data-Driven Approach" Teomara Rutherford PI, Tiffany Barnes & Collin F. Lynch Co-PIs.

7. REFERENCES

[1] V. Aleven, E. Stahl, S. Schworm, F. Fischer, and R. Wallace. Help seeking and help design in interactive learning environments. *Review of Educational Research*, 73(3):277–320, 2003.

[2] A. Bandura. Perceived self-efficacy in cognitive development and functioning. *Educational Psychologist*, 28:117–148.

[3] R. Bartle. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, 1(1):19, 1996.

[4] A. Boyce, K. Doran, A. Campbell, S. Pickford, D. Culler, and T. Barnes. Beadloom game: Adding competitive, user generated, and social features to increase motivation. In *the 6th International Conference on Foundations of Digital Games*, pages 139–146. ACM, 2011.

[5] C. Burgers, A. Eden, M. D. van Engelenburg, and S. Buningh. How feedback boosts motivation and play in a brain-training game. *Computers in Human Behavior*, 48:94–103, 2015.

[6] S. L. Calvert, B. L. Strong, and L. Gallagher. Control as an engagement feature for young children's attention to and learning of computer content. *American Behavioral Scientist*, 48(5):578–589, 2005.

[7] D. B. Clark, B. C. Nelson, H. Y. Chang, M. Martinez-Garza, K. Slack, and C. M. D'Angelo. Exploring newtonian mechanics in a conceptually-integrated digital game: Comparison of learning and affective outcomes for students in taiwan and the united states. *Computers Education*, 57(3):2178–2195, 2011.

[8] D. I. Cordova and M. R. Lepper. Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology*, 88(4):715, 1996.

[9] J. P. Gee. *What video games have to teach us about learning and literacy.* St. Martin's Griffin - Macmillan, New York, USA, 2007.

[10] G. A. Gunter, R. F. Kenny, and E. H. Vick. Taking educational games seriously: Using the retain model to design endogenous fantasy into standalone educational games. *Educational Technology Research and Development*, 56(5-6):511–537, 2008.

[11] J. D. Marx and K. Cummings. Normalized change. *American Journal of Physics*, 75(1):87–91, 2007.

[12] J. Mostow, J. Beck, R. Chalasani, A. Cuneo, P. Jia, and K. Kadaru. A la recherche du temps perdu, or as time goes by: Where does the time go in a reading tutor that listens? In *In International Conference on Intelligent Tutoring Systems*, pages 320–329, 2002.

[13] F. Pajares. Self-efficacy beliefs in academic setting. *Review of Educational Research*, 66:543–578.

[14] J. L. Plass, P. A. O'keefe, B. D. Homer, J. Case, E. O. Hayward, M. Stein, and K. Perlin. The impact of individual, competitive, and collaborative mathematics game play on learning, performance, and motivation. *Journal of Educational Psychology*, 105(4):1050, 2013.

[15] M. Prensky. Computer games and learning: Digital game-based learning. In *Handbook of computer games studies*. The MIT Press, Cambridge, MA, USA, 2005.

[16] T. Rutherford, G. Farkas, G. Duncan, M. Burchinal, M. Kibrick, J. Graham, L. Richland, N. Tran, S. Schneider, L. Duran, and M. Martinez. A randomized trial of an elementary school mathematics software intervention: spatial-temporal math. *Journal of Research on Educational Effectiveness*, 7(4):358–383, 2014.

[17] J. L. Sabourin, J. P. Rowe, B. W. Mott, and J. C. Lester. Considering alternate futures to classify off-task behavior as emotion self-regulation: A supervised learning approach. *JEDM-Journal of Educational Data Mining*, 5(1):9–38, 2013.

[18] S. Thomas, G. Schott, and M. Kambouri. Designing for learning or designing for fun? setting usability guidelines for mobile educational games. *Learning with mobile devices: A book of papers*, pages 173–181, 2004.

Grade Prediction with Temporal Course-wise Influence

Zhiyun Ren
Computer Science
George Mason University
4400 University Drive,
Fairfax, VA 22030
zren4@gmu.edu

Xia Ning
Computer & Information Science
Indiana University - Purdue University Indianapolis
420 University Blvd,
Indianapolis, IN 46202
xning@cs.iupui.edu

Huzefa Rangwala
Computer Science
George Mason University
4400 University Drive,
Fairfax, VA 22030
rangwala@cs.gmu.edu

ABSTRACT

There is a critical need to develop new educational technology applications that analyze the data collected by universities to ensure that students graduate in a timely fashion (4 to 6 years); and they are well prepared for jobs in their respective fields of study. In this paper, we present a novel approach for analyzing historical educational records from a large, public university to perform next-term grade prediction; i.e., to estimate the grades that a student will get in a course that he/she will enroll in the next term. Accurate next-term grade prediction holds the promise for better student degree planning, personalized advising and automated interventions to ensure that students stay on track in their chosen degree program and graduate on time. We present a factorization-based approach called Matrix Factorization with Temporal Course-wise Influence that incorporates course-wise influence effects and temporal effects for grade prediction. In this model, students and courses are represented in a latent "knowledge" space. The grade of a student on a course is modeled as the similarity of their latent representation in the "knowledge" space. Course-wise influence is considered as an additional factor in the grade prediction. Our experimental results show that the proposed method outperforms several baseline approaches and infer meaningful patterns between pairs of courses within academic programs.

Keywords

next-term grade prediction, course-wise influence, temporal effect, latent factor

1. INTRODUCTION

Data analytics is at the forefront of innovation in several of today's popular Educational Technologies (EdTech) [17]. Currently, one of the grand challenges facing higher education is the problem of student retention and graduation [19]. There is a critical need to develop new EdTech applications that analyze the data collected by universities to ensure that students graduate in a timely fashion (4 to 6 years), and they are well prepared for jobs in their respective fields of study. To this end, several universities deploy a suite of software and tools. For example, *degree planners* [1] assist students in deciding their majors or fields of study, choosing the sequence of courses within their chosen major and providing advice for achieving career and learning objectives. *Early warning systems* [27] inform advisors/students of progress, and additionally provide cues for intervention when students are at the risk of failing one or more courses and dropping out of their program of study. In this work, we focus on the problem of next-term grade prediction where the goal is to predict the grade that a student is expected to obtain in a course that he/she may enroll in the next term (future).

In the past few years, several algorithms have been developed to analyze educational data, including Matrix Factorization (MF) algorithms inspired from recommender system research. MF methods decompose the student-course (or student-task) grade matrix into two low-rank matrices, and then the prediction of the grade for a student on an untaken course is calculated as the product of the corresponding vectors in the two decomposed matrices [22, 11]. Traditional MF algorithms have shown a strong ability to deal with sparse datasets [14] and their extensions have incorporated temporal and dynamic information [12]. In our setting, we consider that a student's knowledge is continuously being enriched while taking a sequence of courses; and it is important to incorporate this dynamic influence of sequential courses within our models. Therefore, we present a novel approach referred as Matrix Factorization with Temporal Course-wise Influence (MFTCI) model to predict next term student grades. MFTCI considers that a student's grade on a certain course is determined by two components: (i) the student's competence with respect to each course's topics, content and requirement, etc., and (ii) student's previous performance over other courses. We performed a comprehensive set of experiments on various datasets. The experimental results show that the proposed method outperforms several state-of-the-art methods. The main contributions of our work in this paper are as follows:

1. We model and incorporate temporal course-wise influence in addition to matrix factorization for grade

[1] http://www.blackboard.com/mobile-learning/planner.aspx

prediction. Our experimental results demonstrate significant improvement from course-wise influence.

2. Our model successfully captures meaningful course-wise influences which correlate to the course content.

3. The learned influences between pairs of courses help in understanding pre-requisite structures within programs and tuning academic program chains.

2. RELATED WORK

Over the past few years, several methods have been developed to model student behavior and academic performance [2, 9], and they gain improvement of learning outcomes [21]. Methods influenced by Recommender System (RS) research [1], including Collaborative Filtering (CF) [18] and Matrix Factorization [13], have attracted increasing attention in educational mining applications which relate to student grade prediction [32] and in-class assessment prediction [8]. Sweeney *et. al.* [31, 30] performed an extensive study of several recommender system approaches including SVD, SVD-kNN and Factorization Machine (FM) to predict next-term grade performance. Inspired by content-based recommendation [20] approaches, Polyzou *et. al.* [23] addressed the future course grade prediction problem with three approaches: course-specific regression, student-specific regression and course-specific matrix factorization. Moreover, neighborhood-based CF approaches [25, 4, 6] predict grades based on the student similarities, i.e., they first identify similar students and use their grades to estimate the grades of the students with similar profiles.

In order to capture the changing of user dynamics over time in RS, various dynamic models have been developed. Many of such models are based on Matrix Factorization and state space models. Sun *et. al.* [28, 29] model user preference change using a state space model on latent user factors, and estimate user factors over time using noncausal Kalman filters. Similarly, Chua *et.al.* [5] apply Linear Dynamical Systems (LDS) on Non-negative Matrix Factorization (NMF) to model user dynamics. Ju *et. al.* [12] encapsulate the temporal relationships within a Non-negative matrix formulation. Zhang *et. al.* [34] learn an explicit transition matrix over the latent factors for each user, and estimate the user and item latent factors and the transition matrices within a Bayesian framework. Other popular methods for dynamic modeling include time-weighting similarity decaying [7], tensor factorization [33] and point processes [16]. The method proposed in this paper tackle the challenges of next-term grade prediction which relates to the evolvement of student knowledge over taking a sequence of courses. Our key contribution involves how we incorporate the temporal course-wise relationships within a MF approach. Additionally, the proposed approach learns pairwise relationships between courses that can help in understanding pre-requisite structures within programs and tuning academic program chains.

3. PRELIMINARIES

3.1 Problem Statement and Notations

Formally, student-course grades will be represented by a series of matrices $\{\mathrm{G}_1, \mathrm{G}_2, ..., \mathrm{G}_T\}$ for T terms. Each row of G_t represents a student, each column of G_t represents a course, and each value in G_t, denoted as $g^t_{s,c}$, represents a grade that student s got on course c in term t ($g^t_{s,c} \in (0, 4]$, $g^t_{s,c} = 0$ indicates that student s did not take the course c in term t. We add a small value to failing grade to distinguish 0 score from such situation.). Student-course grades up to the t_{th} term will be represented by $\mathrm{G}^t = \sum_{i=1}^{t} \mathrm{G}_i$ with size of $n \times m$, where n is the number of students and m is the number of courses. Given the database of (student, course, grade) up to term $(T-1)$ (i.e., G^{T-1}), the next-term grade prediction problem is to predict grades for each student on courses they might enroll in the next term T. To simplify the notations, if not specifically stated in this paper, we will use $g_{s,c}$ to denote $g^t_{s,c}$. Our testing set is then (student, course, grade) triples in the T_{th} term, represented by matrix G_T. Rows from the grade matrices representing a student s will simply be represented as $G(s,:)$ and the specific courses that student has a grade for in this row can be given by $c' \in G(s,:)$.

In this paper, all vectors (e.g., $\mathbf{u}_s^\mathsf{T}$ and $\mathbf{v}_c$) are represented by bold lower-case letters and all matrices (e.g., A) are represented by upper-case letters. Column vectors are represented by having the transpose supscript$^\mathsf{T}$, otherwise by default they are row vectors. A predicted/approximated value is denoted by having a $\tilde{}$ head.

4. METHODS

4.1 MF with Temporal Course-wise Influence

We consider the student s' grade on a certain course c, denoted as $g_{s,c}$, as determined by two factors. The first factor is the student s' competence with respect to the course c's topics, content and requirement. This is modeled through a latent factor model, in which s' competence is captured using a size-k latent factor $\mathbf{u}_s$, c's topics and contents are captured using a size-k latent factor $\mathbf{v}_c$ in the same latent space as $\mathbf{u}_s$. Then the competence of s over c is modeled by the "similarity" between $\mathbf{u}_s$ and $\mathbf{v}_c$ via their dot product (i.e., $\mathbf{u}_s^\mathsf{T}\mathbf{v}_c$).

The second factor is the previous performance of student s over other courses. We hypothesize that if course c' has a positive influence on course c, and student s achieved a high grade on c', then s tends to have a high grade on c. Under this hypothesis, we model this second factor as a product between the performance of student on a previous "related" course where the pairwise course relationships are learned in our formulation. Note that we consider this pairwise course influence as time independent, i.e., the influence of one course over another does not change over time. However, the impact from previous performance/grades can be modeled using a decay function over time. Taking these two factors, the estimated grade is given as follows:

$$\begin{aligned}\tilde{g}_{s,c} = \mathbf{u}_s^\mathsf{T}\mathbf{v}_c \\ + \underbrace{e^{-\alpha}\frac{\sum_{c' \in G_{T-1}(s,:)} A(c',c) g_{s,c'}}{|G_{T-1}(s,:)|}}_{\Delta(T-1)} \\ + \underbrace{e^{-2\alpha}\frac{\sum_{c'' \in G_{T-2}(s,:)} A(c'',c) g_{s,c''}}{|G_{T-2}(s,:)|}}_{\Delta(T-2)},\end{aligned} \tag{1}$$

in which $A(c', c)$ is the influence of c' on c, $G_{T-1}(s,:)/G_{T-2}(s,:)$ is the subset of courses out of all courses that s has taken in the first/second previous terms, $|G_{T-1}(s,:)|/|G_{T-2}(s,:)|$ is the number of such taken courses. $e^{-\alpha}/e^{-2\alpha}$ denote the time-decay factors. In Equation 1, we consider previous two terms. More previous terms can be included with even stronger time-decay factors. Given the grade estimation as in Equation 1, we formulate the grade prediction problem for term T as the following optimization problem,

$$\begin{aligned}\min_{U,V,A} \ & \frac{1}{2}\sum_{s,c}(g_{s,c} - \tilde{g}_{s,c})^2 + \frac{\gamma}{2}(\|U\|_F^2 + \|V\|_F^2) \\ & + \tau\|A\|_* + \lambda\|A\|_{\ell_1} \\ & \text{s.t., } A \geq 0\end{aligned}$$

where U and V are the latent non-negative student factors and course factors, respectively; $\|A\|_*$ is the nuclear norm of A, which will induce an A of low rank; and $\|A\|_{\ell_1}$ is the ℓ_1 norm of A, which will introduce sparsity in A. In addition, the non-negativity constraint on A is to enforce only positive influence across courses.

4.1.1 Optimization Algorithm of MFTCI

We apply the ADMM [3] technique for Equation 2 by reformulating the optimization problem as follows,

$$\begin{aligned}\min_{U,V,A,U_1,U_2,Z_1,Z_2} \quad & \frac{1}{2}\sum_{s,c}(g_{s,c} - \tilde{g}_{s,c})^2 + \frac{\gamma}{2}(\|U\|_F^2 + \|V\|_F^2) \\ & +\tau\|Z_1\|_* + \lambda\|Z_2\|_{\ell_1} \\ & +\frac{\rho}{2}(\|A - Z_1\|_F^2 + \|A - Z_2\|_F^2) \\ & +\rho(tr(U_1^\mathsf{T}(A - Z_1))) \\ & +\rho(tr(U_2^\mathsf{T}(A - Z_2))) \\ \text{s.t.,} \quad & A \geq 0\end{aligned}$$

where Z_1 and Z_2 are two auxiliary variables, and U_1 and U_2 are two dual variables. All the variables are solved via an alternating approach as follows.

Step 1: Update U and V. Fixing all the other variables and solving for U and V, the problem becomes a classical matrix factorization problem:

$$\min_{U,V} \frac{1}{2}\sum_{s,c}(f_{s,c} - \mathbf{u}_s^\mathsf{T}\mathbf{v}_c)^2 + \frac{\gamma}{2}(\sum_s \|u_s\|_2^2 + \sum_c \|v_c\|_2^2) \quad (2)$$

where $f_{s,c} = g_{s,c} - \Delta(T-1) - \Delta(T-2)$ (See Eq 1). The matrix factorization problem can be solved using alternating minimization.

Step 2: Update A. Fixing all the other variables and solving for A, the problem becomes

$$\begin{aligned}\min_A \quad & \frac{1}{2}\sum_{s,c}(g_{s,c} - \tilde{g}_{s,c})^2 + \frac{\rho}{2}(\|A - Z_1\|_F^2 + \|A - Z_2\|_F^2) \\ & +\rho(tr(U_1^\mathsf{T}(A - Z_1))) + \rho(tr(U_2^\mathsf{T}(A - Z_2))) \\ \text{s.t.,} \quad & A \geq 0\end{aligned}$$

Using the gradient descent, the elements in A can be updated as follows.

$$\begin{aligned}A(c_i, c_j) = A(c_i, c_j) - lr \times & [\rho(A(c_i, c_j) - Z_1(c_i, c_j)) \\ & + \rho(A(c_i, c_j) - Z_2(c_i, c_j)) + \rho U_1(c_i, c_j) + \rho U_2(c_i, c_j) \\ & - \sum_{s,c_j}(g_{s,c_j} - \tilde{g}_{s,c_j}) \\ & \times \begin{cases} \frac{e^{-\alpha}}{|G_{T-1}(s,:)|} g_{s,c_i} & \text{(if } c_i \text{ is taken in term } T-1\text{)} \\ \frac{e^{-2\alpha}}{|G_{T-2}(s,:)|} g_{s,c_i} & \text{(if } c_i \text{ is taken in term } T-2\text{)}] \end{cases}\end{aligned} \quad (3)$$

with projection into $[0, +\infty)$, where lr is a learning rate.

Step 3: Update Z_1 and Z_2. For Z_1, the problem becomes

$$\min_{Z_1} \tau\|Z_1\|_* + \frac{\rho}{2}\|A - Z_1\|_F^2 + \rho(tr(U_1^\mathsf{T}(A - Z_1))) \quad (4)$$

The closed-form solution of this problem is

$$Z_1 = S_{\frac{\tau}{\rho}}(A + U_1) \quad (5)$$

where $S_\alpha(X)$ is a soft-thresholding function that shrinks the singular values of X with a threshold α, that is,

$$S_\alpha(X) = U\text{diag}((\Sigma - \alpha)_+)V^\mathsf{T} \quad (6)$$

where $X = U\Sigma V^\mathsf{T}$ is the singular value decomposition of X, and

$$(x)_+ = \max(x, 0). \quad (7)$$

For Z_2, the problem becomes

$$\min_{Z_2} \lambda\|Z_2\|_{\ell_1} + \frac{\rho}{2}\|A - Z_2\|_F^2 + \rho(tr(U_2^\mathsf{T})(A - Z_2)) \quad (8)$$

The closed-form solution is

$$Z_2 = E_{\frac{\lambda}{\rho}}(A + U_2) \quad (9)$$

where $E_\alpha(X)$ is a soft-thresholding function that shrinks the values in X with a threshold α, that is,

$$E_\alpha(X) = (X - \alpha, 0)_+ \quad (10)$$

where $()_+$ is defined as in Equation 7.

Step 4: Update U_1 and U_2. U_1 and U_2 are updated based on standard ADMM updates:

$$U_1 = U_1 + (A - Z_1); \qquad U_2 = U_2 + (A - Z_2) \quad (11)$$

In addition, we conduct computational complexity analysis of MFTCI and put it in Appendix.

5. EXPERIMENTS

5.1 Dataset Description

We evaluated our method on student grade records obtained from George Mason University (GMU) from Fall 2009 to Spring 2016. This period included data for 23,013 transfer students and 20,086 first-time freshmen (non-transfer i.e., students who begin their study at GMU) across 151 majors enrolled in 4,654 courses.

Specifically, we extracted data for six large and diverse majors for both non-transfer and transfer students. These majors include: (i) Applied Information Technology (AIT), (ii)

Table 1: Dataset Descriptions

Major	Non-Transfer Students			Transfer Students		
	#S	#C	#(S,C)	#S	#C	#(S,C)
AIT	239	453	5,739	982	465	14,396
BIOL	1,448	990	33,527	1,330	833	22,691
CEIE	393	642	9,812	227	305	4,538
CPE	340	649	7,710	91	219	1,614
CS	908	818	18,376	480	464	7,967
PSYC	911	874	22,598	1504	788	24,661
Total	4,239	1,115	97,762	4,614	1,019	75,867

#S, #C and #S-C are number of students, courses and student-course pairs in educational records across the 6 majors from Fall 2009 to Spring 2016, respectively.

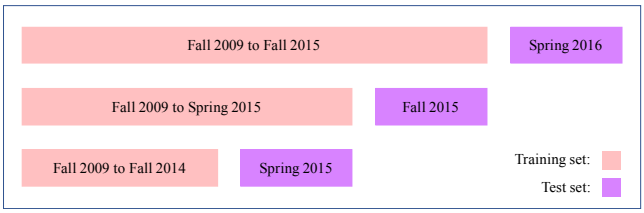


Figure 1: Different Experimental Protocols

Biology (BIOL), (iii) Civil, Environmental and Infrastructure Engineering (CEIE), (iv) Computer Engineering (CPE) (v) Computer Science (CS) and (vi) Psychology (PSYC). Table 1 provides more information about these datasets.

5.2 Experimental Protocol

To assess the performance of our next-term grade prediction models, we trained our models on data up to term $T-1$ and make predictions for term T. We evaluate our method for three test terms, i.e., Spring 2016, Fall 2015 and Spring 2015. As an example, for evaluating predictions for term Fall 2015, data from Fall 2009 to Spring 2015 is considered as training data and data from Fall 2015 is testing data. datasets. Figure 1 shows the three different train-test splits.

5.3 Evaluation Metrics

We use **Root Mean Squared Error (RMSE)** and **Mean Absolute Error (MAE)** as metrics for evaluation, and are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{s,c \in G_T} (g_{s,c} - \tilde{g}_{s,c})^2}{|G_T|}},$$

$$MAE = \frac{\sum_{s,c \in G_T} |g_{s,c} - \tilde{g}_{s,c}|}{|G_T|}$$

where $g_{s,c}$ and $\tilde{g}_{s,c}$ are the ground truth and predicted grade for student s on course c, and G_T is the testing set of (student, course, grade) triples in the T_{th} term. Normally, in next-term grade prediction problem, MAE is more intuitive than RMSE since MAE is a straightforward method which calculates the deviation of errors directly while RMSE has implications such as penalizing large errors more.

For our dataset, a student's grade can be a letter grade (i.e. A, A-, . . . , F). As done previously by Polyzou et. al. [24] we define a tick to denote the difference between two consecutive letter grades (e.g., C+ vs C or C vs C-). To assess the performance of our grade prediction method, we convert the predicted grades into their closest letter grades and compute the percentage of predicted grades with no error (or 0-ticks), within 1-tick and within 2-ticks denoted by Pct_0, Pct_1 and Pct_2, respectively. For the problem of course selection and degree planning, courses predicted within 2 ticks can be considered sufficiently correct. We name these metrics as **Percentage of Tick Accuracy (PTA)**.

5.4 Baseline Methods

We compare the performance of our proposed method to the following baseline approaches.

5.4.1 *Matrix Factorization*

Matrix factorization is known to be successful in predicting ratings accurately in recommender systems [26]. This approach can be applied directly on next-term grade prediction problem by considering student-course grade matrix as a user-item rating matrix in recommender systems. Based on the assumption that each course and student can be represented in the same low-dimensional space, corresponding to the knowledge space, two low-rank matrices containing latent factors are learned to represent courses and students [30]. Specifically, the grade a student s will achieve on a course c is predicted as follows:

$$\tilde{g}_{s,c} = \mu + \mathbf{p}_s + \mathbf{q}_c + \mathbf{u}_s^\top \mathbf{v}_c \quad (12)$$

where μ is a global bias term, $\mathbf{p}_s$ ($\mathbf{p} \in \mathbb{R}^n$) and $\mathbf{q}_c$ ($\mathbf{q} \in \mathbb{R}^m$) are the student and course bias terms (in this case, for student s and course c), respectively, and $\mathbf{u}_s$ ($\mathbf{U} \in \mathbb{R}^{k \times n}$) and $\mathbf{v}_c$ ($\mathbf{V} \in \mathbb{R}^{k \times m}$) are the latent factors for student s and course c, respectively.

5.4.2 *Matrix Factorization without Bias (MF_0)*

We only considered the student and course latent factors to predict the next-term grades. Therefore, the grade a student s will achieve on a course c is calculated as follows:

$$\tilde{g}_{s,c} = \mathbf{u}_s^\top \mathbf{v}_c \quad (13)$$

5.4.3 *Non-negative Matrix Factorization (NMF) [15]*

We add non-negative constraints on matrix $\mathbf{U}$ and matrix $\mathbf{V}$ in Equation 13. The non-negativity constraints allows MF approaches to have better interpretability and accuracy for non-negative data [10].

6. RESULTS AND DISCUSSION

6.1 Overall Performance

Table 2 presents the comparison of Pct_0, Pct_1 and Pct_2 for non-transfer students for the three terms considered as test: Spring 2016, Fall 2015 and Spring 2015. We observe that the MFTCI model outperforms the baselines across the different test sets. On average, MFTCI outperforms the MF, MF_0 and NMF methods by 34.18%, 11.59% and 4.08% in terms of Pct_0, 16.64%, 7.96% and 4.03% in terms of Pct_1, and 2.10%, 3.00% and 1.98% in terms of Pct_2, respectively. We observe similar results for transfer students as well (not included here for brevity).

Table 2: Comparison Performance with PTA (%)

Methods	Spring 2016			Fall 2015			Spring 2015		
	$Pct_0(\uparrow)$	$Pct_1(\uparrow)$	$Pct_2(\uparrow)$	Pct_0	Pct_1	Pct_2	Pct_0	Pct_1	Pct_2
MF	13.25	27.71	58.02	12.05	26.63	58.89	13.03	26.09	54.83
MF_0	16.52	31.65	57.46	15.51	30.03	55.64	15.53	29.53	54.94
NMF	13.21	27.04	57.18	15.33	30.12	56.15	15.56	29.23	54.93
MFTCI	**19.78**	**35.52**	**61.44**	**19.71**	**35.16**	**60.12**	**18.56**	**32.78**	**58.80**

i) "↑" indicates the higher the better. ii) Reported values of Pct_0, Pct_1 and Pct_2 are percentages. iii) Best performing methods are highlighted with bold.

Table 3 presents the performance of the baselines and MFTCI model for the three different terms of both non-transfer and transfer students using RMSE and MAE as evaluation metrics. The MFTCI model consistently outperforms the baselines across the different datasets in terms of MAE. In addition, the results shows that MF_0, NMF and MFTCI tend to have better performance for Spring 2016 term than Fall 2015 term. Similar trend is observed between Fall 2015 term and Spring 2015 term. This suggests that MFTCI is likely to have better performance with more information in the training set.

6.2 Analysis on Individual Majors

We divide non-transfer students based on their majors and test the baselines and MFTCI model on each major, separately. Table 4 shows the comparison of Pct_0, Pct_1 and Pct_2 on different majors. The results show that MFTCI has the best performance for almost all the majors. Among all the results, MFTCI has the highest accuracy when predicting grades for PSYC and BIOL students for which we have more student-course pairs in the training set.

6.3 Effects from Previous Terms on MFTCI

In order to see the influence of number of previous terms considered in MFTCI, we run our model with only $\Delta(T-1)$ in Equation 1. This method is represented as $MFTCI_{p1}$. Figure 2 shows the comparison results of MAE for six subsets of data which are reported in Table 3, where "NTR" stands for non-transfer students and "TR" stands for transfer students. The results show that MFTCI consistently outperforms $MFTCI_{p1}$ on all datasets. This suggests that considering two previous terms is necessary for achieving good prediciton results. Moreover, since we consider that the student's knowledge is modeled using an exponential decaying function over time, we do not include the influence from the third previous term in our model as its influence for the grade prediction is negligible in comparison to the previous two terms.

6.4 Visualization of Course Influence

To interpret what is captured in the course influence matrix A (See Eq 1), we extract the top 20 values with the corresponding course names (and topics) for analysis. Figure 3 and 4 show the captured pairwise course influences for CS and AIT majors, respectively. Each node corresponds to one course which is represented by the shortened course's name. We can notice from the figures that most influences reflect content dependency between courses. For example, in the CS major, "Object Oriented Programming" course has significant influence on performance of "Low-Level Programming" course (the former one is also the latter one's prerequisite course); "Linear Algebra" and "Discrete Mathematics" have influence on each other; "Formal Methods & Models" course has influence on "Analysis of Algorithms" course. In case of the AIT major, both "Introductory IT" course and "Introductory Computing" course have influence on "IT Problem & Programming" course; "Multimedia & Web Design" course has influence on both "Applied IT Programming" course and "IT in the Global Economy" course. GMU has a sample schedule of eight-term courses for each major in order to guide undergraduate students to finish their study step by step based on the level, content and difficulty of courses [2]. Among the identified relationships shown in Figures 3 and 4 we found 17 and 13 of the CS and AIT courses influences in the guide map, respectively. The rest of the identified influences are among other general electives but required courses (e.g., "Public Speaking" course), or specific electives pertaining to the major (e.g., "Research Methods" course). This shows that our model learns meaningful course-wise influences and successfully uses it to improve MF model.

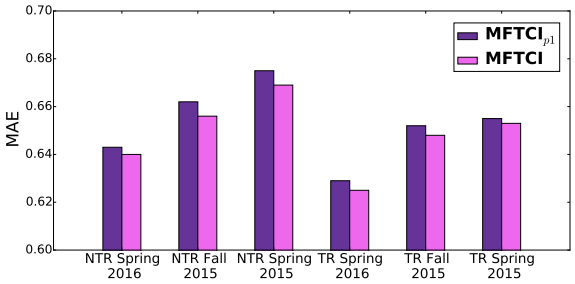


Figure 2: Comparison performance for $MFTCI_{p1}$ and MFTCI

Figure 5 shows the identified course influences for the BIOL, CEIE, CPE and PSYC majors. These identified course-wise influences seem to capture similarity of course content.

7. CONCLUSION AND FUTURE WORK

We presented a Matrix Factorization with Temporal Course-wise Influence (MFTCI) model that integrates factorization models and the influence of courses taken in the preceding terms to predict student grades for the next term.

We evaluate our model on the student educational records from Fall 2009 to Spring 2016 collected from George Ma-

[2]http://catalog.gmu.edu

Table 3: Comparison Performance with RMSE and MAE.

Methods	Non-Transfer Students						Transfer Students					
	Spring 2016		Fall 2015		Spring 2015		Spring 2016		Fall 2015		Spring 2015	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
MF	0.999	0.754	1.037	0.786	1.023	0.784	0.925	0.688	**0.921**	0.686	0.985	0.732
MF_0	0.929	0.714	0.977	0.752	1.014	0.778	0.893	0.668	0.944	0.705	1.011	0.765
NMF	1.020	0.769	**0.967**	0.746	**1.000**	0.771	0.906	0.683	0.932	0.701	**0.979**	0.746
MFTCI	**0.928**	**0.685**	0.982	**0.717**	1.012	**0.750**	**0.887**	**0.636**	0.927	**0.662**	1.000	**0.721**

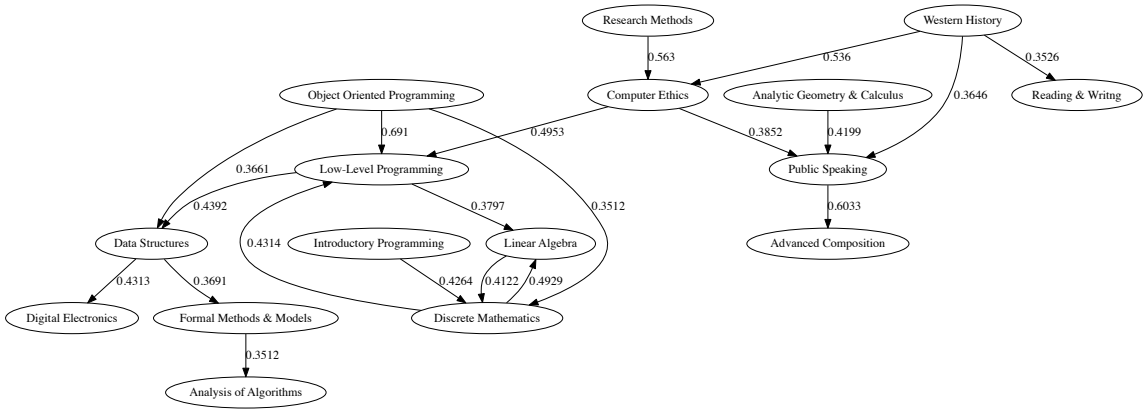


Figure 3: Identified course influences for CS major

Table 4: Comparison Performance for Different Majors

	Methods	AIT	BIOL	CEIE	CPE	CS	PSYC
Pct_0	MF	18.71	18.00	15.99	12.99	15.98	20.18
	MF_0	19.45	22.10	16.70	14.21	16.47	22.12
	NMF	19.77	22.16	**17.01**	14.32	16.61	22.17
	MFTCI	**22.30**	**24.24**	16.80	**14.32**	**17.32**	**25.83**
Pct_1	MF	37.95	35.43	31.47	27.86	31.53	39.41
	MF_0	37.21	39.68	31.87	**27.97**	30.51	39.63
	NMF	36.79	39.74	31.67	27.19	30.43	39.36
	MFTCI	**39.64**	**40.87**	**32.38**	27.53	**31.78**	**42.29**
Pct_2	MF	**67.02**	67.78	58.66	52.28	56.91	**71.01**
	MF_0	66.17	67.54	58.35	50.72	56.24	67.74
	NMF	66.70	67.54	58.55	51.17	56.17	67.79
	MFTCI	66.70	**68.25**	**58.76**	**52.94**	**58.18**	68.29

son University. The dataset in this study contains both non-transfer and transfer students from six different majors. Our experimental evaluation shows that MFTCI consistently outperforms the different state-of-the-art methods. Moreover, we analyze the effects from previous terms on MFTCI, and we make the conclusion that it is necessary to consider two previous terms. In addition, we visualize the patterns learned between pairs of courses. The results strongly demonstrate that the learned course influences correlate with the course content within academic programs.

In the future, we will explore incorporation of additional constraints over the the pairwise course influence matrix, such as prerequisite information, compulsory and elective provision of a course. We will explore using the course influence information to build a degree planner for future students.

8. ACKNOWLEDGMENTS

Funding was provided by NSF Grant, 1447489.

APPENDIX

A. COMPUTATIONAL COMPLEXITY ANALYSIS

The computational complexity of MFTCI is determined by the four steps in the alternating approach as described above. To update U and V as in Equation 2 using gradient descent method via alternating minimization, the computational complexity is $O(\text{niter}_{uv}(k \times n_{s,c} + k \times m + k \times n)) = O(\text{niter}_{uv}(k \times n_{s,c}))$ (typically $n_{s,c} \geq \max(m, n)$), where $n_{s,c}$ is the total number of student-course dyads, n is the number of students, m is the number of courses, k is the latent dimensions of U and V, and niter_{uv} is the number of iterations. To update A as in Equation 3 using gradient descent method, the computational complexity is upper-bounded by $O(\text{niter}_a(n_{cc} \times \frac{n_{s,c}}{m}))$, where n_{cc} is the number of course pairs that have been taken by at least one student, $\frac{n_{s,c}}{m}$ is the average number of students for a course, which upper bounds the average number of students who co-take two courses, and niter_a is the number of iteractions. Essentially, to update A, we only need to update $A(c_i, c_j)$ where c_i and c_j have been co-taken by some students. For $A(c_i, c_j)$ where c_i and c_j have never been taken together, they will remain 0. To update Z_1 as in Equation 4, a singular value decomposition is involved and thus its computational complexity is upper bounded by $O(m^3)$. To update Z_2 as in Equation 8, the computational complexity is $O(m^2)$. To update

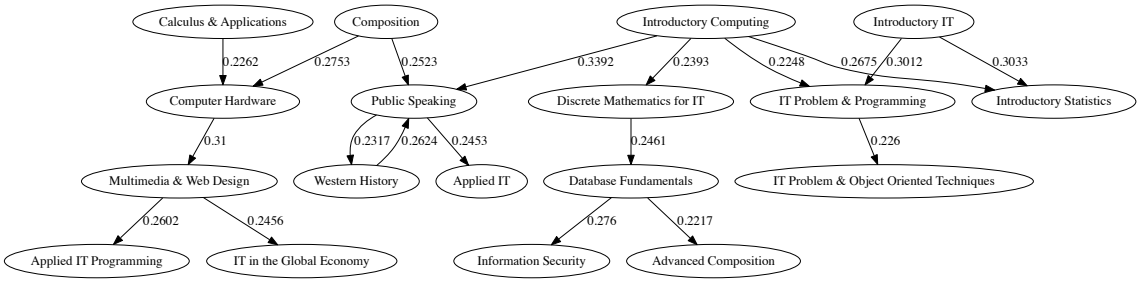
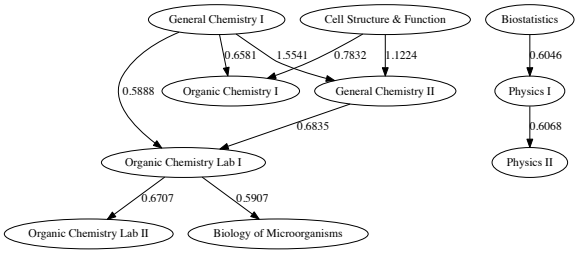
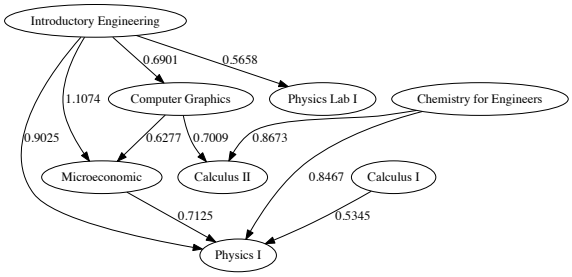


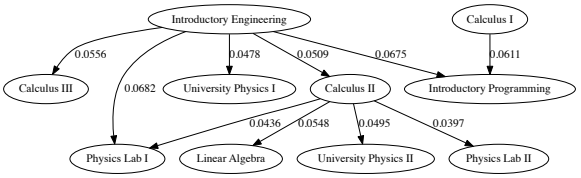
Figure 4: Identified course influences for AIT major



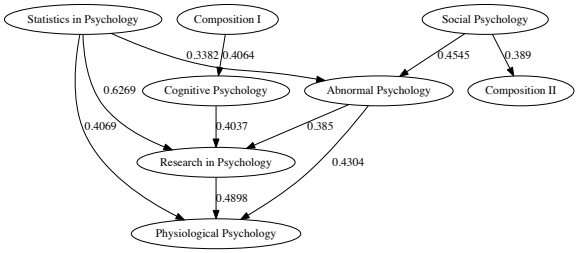
(a) Identified course influences for BIOL major



(b) Identified course influences for CEIE major



(c) Identified course influences for CPE major



(d) Identified course influences for PSYC major

Figure 5: Identified course influences for different majors

U_1 and U_2 as in Equation 11, the computational complexity is $O(m^2)$. Thus, the computational complexity for MTFCI is $O(\text{niter}(\text{niter}_{uv}(k \times n_{s,c}) + \text{niter}_a(n_{cc} \times \frac{n_{s,c}}{m}) + m^3 + m^2))$ $= O(\text{niter}(\text{niter}_{uv}(k \times n_{s,c}) + \text{niter}_a(n_{cc} \times \frac{n_{s,c}}{m}) + m^3))$, where niter is the number of iterations for the four steps. Although the complexity is dominated by m^3 due to the SVD on $A + U_1$, since n (i.e., the number of courses) is typically not large, the run time will be more dominated by $n_{s,c}$ (i.e., the number of student-course dyads).

B. REFERENCES

[1] Charu C. Aggarwal. *Recommender Systems: The Textbook*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[2] RSJD Baker et al. Data mining for education. *International encyclopedia of education*, 7:112–118, 2010.

[3] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[4] Hana Bydžovská. Are collaborative filtering methods suitable for student performance prediction? In *Portuguese Conference on Artificial Intelligence*, pages 425–430. Springer, 2015.

[5] Freddy Chong Tat Chua, Richard J Oentaryo, and Ee-Peng Lim. Modeling temporal adoptions using dynamic matrix factorization. In *2013 IEEE 13th International Conference on Data Mining*, pages 91–100. IEEE, 2013.

[6] Tristan Denley. Course recommendation system and method, January 10 2013. US Patent App. 13/441,063.

[7] Yi Ding and Xue Li. Time weight collaborative filtering. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 485–492, New York, NY, USA, 2005. ACM.

[8] Asmaa Elbadrawy, Scott Studham, and George Karypis. Personalized multi-regression models for predicting students performance in course activities. *UMN CS*, pages 14–011, 2014.

[9] Wu He. Examining studentsâĂŹ online interaction in a live video streaming environment using data mining and text mining. *Computers in Human Behavior*, 29(1):90–102, 2013.

[10] Ngoc-Diep Ho. *Nonnegative matrix factorization algorithms and applications*. PhD thesis, ÉCOLE POLYTECHNIQUE, 2008.

[11] Chein-Shung Hwang and Yi-Ching Su. Unified clustering locality preserving matrix factorization for student performance prediction. *IAENG Int. J. Comput. Sci*, 42(3):245–253, 2015.

[12] Bin Ju, Yuntao Qian, Minchao Ye, Rong Ni, and Chenxi Zhu. Using dynamic multi-task non-negative matrix factorization to detect the evolution of user preferences in collaborative filtering. *PloS one*, 10(8):e0135090, 2015.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[14] Yehuda Koren, Robert Bell, Chris Volinsky, et al. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[15] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[16] Dixin Luo, Hongteng Xu, Yi Zhen, Xia Ning, Hongyuan Zha, Xiaokang Yang, and Wenjun Zhang. Multi-task multi-dimensional hawkes processes for modeling event sequences. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 3685–3691. AAAI Press, 2015.

[17] Rabab Naqvi. Data mining in educational settings. *Pakistan Journal of Engineering, Technology & Science*, 4(2), 2015.

[18] Xia Ning, Christian Desrosiers, and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 37–76. Springer, 2015.

[19] Michelle Parker. Advising for retention and graduation. 2015.

[20] Michael J. Pazzani and Daniel Billsus. The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.

[21] Alejandro Peña-Ayala. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert systems with applications*, 41(4):1432–1462, 2014.

[22] Štefan Pero and Tomáš Horváth. Comparison of collaborative-filtering techniques for small-scale student performance prediction task. In *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, pages 111–116. Springer, 2015.

[23] Agoritsa Polyzou and George Karypis. Grade prediction with models specific to students and courses. *International Journal of Data Science and Analytics*, pages 1–13, 2016.

[24] Agoritsa Polyzou and George Karypis. Grade prediction with models specific to students and courses. *International Journal of Data Science and Analytics*, pages 1–13, 2016.

[25] Sanjog Ray and Anuj Sharma. A collaborative filtering based approach for recommending elective courses. In *International Conference on Information Intelligence, Systems, Technology and Management*, pages 330–339. Springer, 2011.

[26] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B Kantor. Recommender systems handbook., 2011.

[27] Jill M Simons. *A National Study of Student Early Alert Models at Four-Year Institutions of Higher Education.* ERIC, 2011.

[28] John Z Sun, Dhruv Parthasarathy, and Kush R Varshney. Collaborative kalman filtering for dynamic matrix factorization. *IEEE Transactions on Signal Processing*, 62(14):3499–3509, 2014.

[29] John Z Sun, Kush R Varshney, and Karthik Subbian. Dynamic matrix factorization: A state space approach. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1897–1900. IEEE, 2012.

[30] Mack Sweeney, Jaime Lester, and Huzefa Rangwala. Next-term student grade prediction. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 970–975. IEEE, 2015.

[31] Mack Sweeney, Huzefa Rangwala, Jaime Lester, and Aditya Johri. Next-term student performance prediction: A recommender systems approach. *arXiv preprint arXiv:1604.01840*, 2016.

[32] Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, and Lars Schmidt-Thieme. Recommender system for predicting student performance. *Procedia Computer Science*, 1(2):2811–2819, 2010.

[33] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G. Carbonell. *Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization*, pages 211–222. 2010.

[34] Chenyi Zhang, Ke Wang, Hongkun Yu, Jianling Sun, and Ee-Peng Lim. Latent factor transition for dynamic collaborative filtering. In *SDM*, pages 452–460. SIAM, 2014.

Toward the Automatic Labeling of Course Questions for Ensuring their Alignment with Learning Outcomes

S. Supraja
Nanyang Technological University
50 Nanyang Ave
Singapore 639798
ssupraja001@e.ntu.edu.sg

Kevin Hartman
Nanyang Technological University
50 Nanyang Ave
Singapore 639798
khartman@ntu.edu.sg

Sivanagaraja Tatinati
Nanyang Technological University
50 Nanyang Ave
Singapore 639798
tatinati@ntu.edu.sg

Andy W. H. Khong
Nanyang Technological University
50 Nanyang Ave
Singapore 639798
andykhong@ntu.edu.sg

ABSTRACT

Expertise in a domain of knowledge is characterized by a greater fluency for solving problems within that domain and a greater facility for transferring the structure of that knowledge to other domains. Deliberate practice and the feedback that takes place during practice activities serve as gateways for developing domain expertise. However, there is a difficulty in consistently aligning feedback about a learner's practice performance with the intended learning outcomes of those activities – especially in situations where the person providing feedback is unfamiliar with the intention of those activities. To address this problem, we propose an intelligent model to automatically label opportunities for practice (assessment questions) according to the learning outcomes intended by the course designers. As a proof of concept, we used a reduced version of Bloom's Taxonomy to define the intended learning outcomes. Using a factorial design, we employed term frequency-inverse document frequency (TF-IDF) and latent Dirichlet allocation (LDA) to transform questions from text to word weightages with support vector machine (SVM) and extreme learning machine (ELM) to train and automatically label the questions. We trained our models with 120 questions labeled by the subject matter expert of an undergraduate engineering course. Compared to existing works which create models based on a self-generated dataset, our proposed approach uses 30 untrained questions from online/textbook sources to validate the performance of our models. Exhaustive comparison analysis of the testing set showed that TF-IDF with ELM outperformed the other combinations by yielding 0.86 reliability (F1 measure) with the subject matter expert.

Keywords

Learning outcomes, Term frequency-inverse document frequency, Latent Dirichlet allocation, Extreme learning machine, Support vector machine

1. INTRODUCTION

Increasingly, modern curriculum design in tertiary and adult learning settings has become a collaborative endeavor between subject matter experts, learning designers, and learning technologists. While these teams employ a variety of process models for the planning, execution, and revision of their curriculum and activity designs, often greater attention is paid to the construction of a course design and the course content rather than the assessment practices that measure learning and their ongoing maintenance.

The algorithms and use case described in this paper exist in a particular context of outcome-based education. In this context, learning is defined by observable changes in a learner's behavior. These changes commensurate with Krathwohl's model of learning objectives [1] but learning outcomes go beyond objectives. Learning outcomes are predicated on having learners observably demonstrate their growing understanding of a topic or proficiency within a field [2]. When learning activities become more open-ended and exploratory, and when learners are offered choices for how to proceed, learners often look to how they will ultimately be assessed to gauge which learning strategies they should employ [3].

When a course's learning activities support its assessment practices and the assessment practices support the types of outcomes that are relevant to learners in the future, the course's activities and intended learning outcomes exhibit constructive alignment with each other [2]. Adhering to constructive alignment creates a seamless path from learning, to applying, to transferring concepts and relationships when solving novel problems.

However, the promise of constructive alignment is not easily delivered upon. Oftentimes, a course's learning outcomes cannot be measured by its assessment practices, or its assessment practices are decontextualized from the types of activities and practices learners are actually preparing for [4]. Whether in the context of higher learning or professional development, when thinking about developing flexible, life-long learners it is paramount to have mechanisms in place to support learners as they work to gain domain expertise. These processes should reliably measure learning and link assessment practices to authentic activities.

1.1 Learning design for domain expertise

Prior work in designing for adaptive domain expertise, the kind of expertise necessary for learners to function in changing environments and flexible job scopes, has shown that learning design teams need to be cognizant of three elements which will be discussed in turn.

1.1.1 Levels of learning outcomes

Learning outcomes range in sophistication and vary by field. In medicine, Miller's Pyramid [5] lists learning outcomes beginning with knowing about a subject, progressing to knowing how to do something, to being able to actually demonstrate it in a contrived setting like a role-play with actors, and to being able to demonstrate it in a real environment like a surgical theater [6]. The idea is based on the belief that the development of expertise is a progression from

the recall of facts to the execution of skills. However, as research on problem based learning has shown, demonstration of skill and the recall of facts can proceed independently of each other depending on the learning environment [7].

In [8], a field agnostic method of classifying learning outcomes based on their quality is presented. Essentially, the Structure of Observed Learning Outcomes (SOLO) taxonomy identifies the level of cognitive sophistication a learning outcome requires. Lower level learning outcomes indicate a learner is capable of remembering facts in isolation. More sophisticated levels require learners to assimilate information from various sources to make connections and transform that understanding into something new.

Perhaps the most popular listing of learning outcomes is Bloom's Taxonomy. Similar to Miller's Pyramid, Bloom's Revised Taxonomy also begins with the retrieval of facts and information as its foundation and builds up to application of knowledge and further to analyzing, evaluating, and creating. Because of its simplicity and familiarity with learning designers and subject matter experts alike, Bloom's Taxonomy can easily be used to identify the levels of learning outcomes in a course [9].

1.1.2 Opportunities for deliberate practice

Along with identifying a learning activity's intended outcomes, expertise development requires opportunities for deliberate practice. In contrast to repetitive practice intended for learners to develop automaticity in either the recall of information or the application of a skill, often during time-limited tasks, deliberate practice focuses on mastering the nuances of the domain itself to fine-tune performance [10]. In fact, a learner's level of grit, a combination of perseverance and passion, predicts how close to expert performance a learner will eventually show [11].

The key difference in processes between repetitive practice and deliberate practice leads to different forms of expertise: adaptive and routine [12]. Routine forms of expertise allow a learner to conduct a task at an optimal level. Adaptive expertise allows learners to learn new tasks or solve novel problems at an accelerated rate. In an industrial setting, routine expertise helps a worker complete a particular job function. Adaptive expertise enables that same worker to retrain to fill new job functions. Typically, the amount of time necessary to achieve expert performance in a domain is in the order of years to decades [13]. However, incremental improvement can be seen in a few practice cycles when activities align to the intended learning outcomes.

1.1.3 Formative assessments and actionable feedback

Hand in hand with creating opportunities for deliberate practice is providing formative feedback to the learner about how to improve that practice while that improvement is still relevant. Imagine students who diligently answer every question in an engineering textbook but never receive feedback on the quality of their solutions. In this case, the learners would be unable to gauge their performance in relation to the course learning outcomes or have an idea about how to improve their performance in the future. Now imagine if those same students do receive feedback, but that feedback arrives after the course's final examination. If the content of the course is mostly self-contained and will not be revisited, the feedback is mostly irrelevant.

Formative feedback consists of two parts: 1) an interpretable indication of a learner's performance on an assessment of learning with respect to a standard of performance (learning outcome) and 2) the opportunity to improve performance before the final evaluation [14].

Cognitive tutors provide a clear example of the power of coupling formative assessment and actionable feedback together in the domain of mathematics learning [15]. By presenting learners with a series of structured problems, cognitive tutors are capable of intervening at any point during the problem-solving process to provide students with feedback about their performance. This feedback may be the identification of an error, the presentation of a hint, or the request for more information about the learner's reasoning. After the feedback, learners have the opportunity to adjust their problem-solving heuristics to improve their performance going forward.

Such an interaction sequence works with highly structured tasks with application-oriented learning outcomes. However, the feedback cycle is more difficult to manage when the learning outcomes are aligned to higher-order reasoning like evaluation, analyzing and creating. These outcomes have multiple paths for reaching a satisfactory answer.

With this difficulty in mind, we looked at techniques to automate the process of identifying the reasoning level of text-based assessment items (questions) with the intention of better aligning questions to learning outcomes as a first step toward being able to provide opportunities for deliberate practice. Subsequently, the outcome of our proposed work is to link actionable feedback to a learner's performance on assessment items.

1.2 Automated question classification techniques

Prior work has shown the viability of automatically labeling questions in accordance with a course's learning outcomes. However, our work goes beyond labeling existing content to helping course instructors promote deliberate practice and expertise development by providing a method of finding new questions that align to the course designer's original intended learning outcomes. We highlight the drawbacks of prior work and how our proposed approach addresses those limitations.

1.2.1 Labeling questions based on difficulty level

Early attempts at automatically labeling questions relied on subject matter experts to pre-define the difficulty levels of questions. Artificial neural network trained by backpropagation then used the question features and assigned difficulty levels in the training set to classify new questions. A five-dimensional feature vector that consisted of query-text relevance, mean term frequency, length of questions and answers, term frequency distribution (variance), distribution of questions and answers in a text were used. The method yielded an F1 measure, a classification reliability metric that measures a test's accuracy, of 0.78 [16]. However, a major pitfall this method is its lack of semantic analysis.

Entropy-Based Decision Tree has also been used to label questions [17]. The weakness in this strategy is that there is high possibility of overfitting the model during the training phase that then negatively affects the subsequent prediction performance.

1.2.2 Labeling questions based on Bloom's Taxonomy using Natural Language Processing

Natural Language Processing (NLP) has been used for the generation of assessments, answering questions, supporting users in Learning Management Systems and preparing course materials. The Wordnet package has been used to detect semantic similarity. By performing a rule-based approach, the accuracy of labeling a

question based on Bloom's Taxonomy reaches 82% [18]. To improve the rule-based approach, a hybrid technique of using an N-gram classifier with a rule-based approach has also been explored. Rules were based on combining parts-of-speech tagging, and the N-gram classifier found the probabilities of predicting certain words. Such a hybrid method yielded an F1 measure of 0.86 [19].

1.2.3 Labeling questions based on Bloom's Taxonomy using machine learning techniques

Machine learning algorithms can be broadly split into either supervised or unsupervised training implementations. Generally, supervised training is adopted when, during training, labels have been pre-determined and questions are labeled by an expert. The most commonly used method in such cases is the term frequency-inverse document frequency (TF-IDF). The algorithm assigns weightages to individual words in a question statement to define a custom vector space to each question.

Machine learning techniques such k-nearest neighbors, Naïve Bayes and support vector machine (SVM) have been implemented for labeling questions. When doing a performance comparison among these three techniques, an F1 measure of 0.71 was achieved using SVM [20]. To increase the accuracy level, additional features were incorporated in future versions of the work. Three different feature selection processes, namely: Odd Ratio, Chi-square statistic and Mutual Information were used with the three machine learning techniques. The F1 measure result reached 0.9 [21].

Furthermore, an integrated approach of feature extraction has been proposed by using headword, semantic, keyword and syntactic extractions, which are fed into SVM [22]. However, this work has not yet been completed by using a testing dataset to quantify the reliability of prediction.

A major downside in existing works is that both the training as well as testing questions are part of the same course curriculum; the questions are generated by the same author/instructor. Even when a high F1 measure is achieved, it does not enable the algorithm to label questions written by another subject matter expert. Our work increases the flexibility of labeling methods by testing our models with a new set of questions compiled from textbook and online resources.

In addition, our work introduces extreme learning machine (ELM), which has been shown to outperform SVM during similar labeling tasks [23]. Moreover, we introduce LDA as an alternative technique to TF-IDF for transforming question statements into numerical word weightages.

By comparing combinations of these new techniques with more traditional techniques, we aim to gauge which combination attains the highest labeling reliability with the subject matter expert when automatically labeling untrained questions. For our purposes, using the combination with the highest F1 measure (fewest false negatives and false positives) becomes paramount. In our use case, a mislabeling by the algorithm will lead to the wrong set of practice questions to be given to students and diminish the impact of deliberate practice on reaching the intended learning outcomes.

2. METHODS

2.1 Materials

2.1.1 Labeling scheme

The core of this study centers on a labeling scheme for identifying the sophistication of learning outcomes based on a simplified version of Bloom's Taxonomy. In this labeling scheme, the first two levels of Bloom's Taxonomy (Remembering and Understanding) were collapsed into Remember. Applying remained its own category. All of the higher-order reasoning categories (Analyzing, Evaluating, and Creating) were collapsed into Transfer. Figure 1 shows how our labeling scheme categories map onto the original categories from Bloom's Revised Taxonomy.

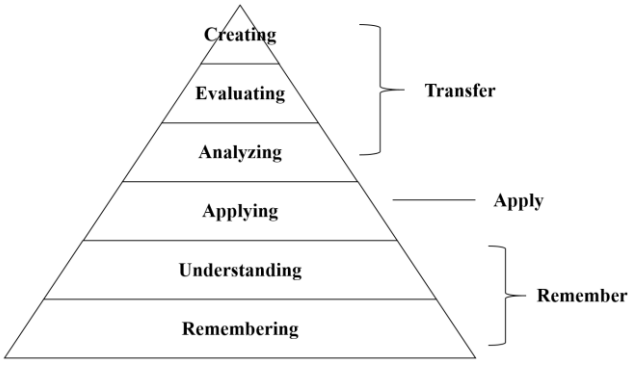


Figure 1: Mapping of Bloom's Revised Taxonomy [24]

We collapsed the taxonomy into three categories for two reasons. First, the subject matter expert tasked with labeling the questions was unsure about how reliably the questions could be labeled by someone without a background in learning design, educational psychology, or curriculum development. Collapsing the categories to Remember, Apply, and Transfer made manually labeling hundreds of questions to train the machine learning algorithms more tractable. Second, collapsing the categories had the effect of making Bloom's Taxonomy more analogous to the successful use cases of Miller's Pyramid by subject matter experts in both higher education and professional development settings [5].

2.1.2 Question dataset

The dataset consists of a total of 150 questions used for training and testing the machine learning algorithms based on the content of an undergraduate electrical and electronic engineering course.

For this study, we formed a training set of 120 questions by randomly selecting 40 Remember, Apply, and Transfer items from the larger question pool of more than 200 questions used in that course. The pool came from a repository of four years' worth of assignment, homework, quiz and exam questions presented to students. These questions prompt students for a range of answer types (i.e., open-ended, multiple-choice, short-structured, essay).

We then created a testing set of 30 new questions compiled from external sources such as textbooks and online question banks. This set was also balanced with equal representation of Remember, Apply, and Transfer questions.

2.2 Data pre-processing procedures

We pre-processed the raw questions in two phases. First, the subject matter expert labeled every question according to the labeling scheme described above. Second, we transformed the text of every question into a machine-readable format before passing them through the machine learning algorithms.

2.2.1 Subject matter expert pre-processing

The subject matter expert manually labeled each question in the training set based on its intended learning outcome (Remember, Apply or Transfer). The subject matter expert then labeled the 30 new questions in the testing set in the same manner. These new questions are labeled for the purpose of knowing the ground truth for performance evaluation. Table 1 below shows some examples of the labeled questions.

Table 1 - Examples of labeled questions

Remember
Consider a signal described by y[n] = 2n +4. What would be the amplitude of the signal at sample index n=3?
Apply
Consider the following input and output signals: find the transfer function and state the poles and zeros of this transfer function.
Transfer
Describe how the bandpass filter can be utilized for radar applications.

2.2.2 Text pre-processing

The text transformation began by excising all equations, mathematical symbols and diagrams from the questions. We only kept the core of the question prompts by removing the descriptive and explanatory text from scenario and hypothetical questions. For example, if a question began by setting the stage with "Peter has been asked to perform…", followed by the question prompt "How much voltage should Peter expect in the circuit?", all of the descriptive text prior to the question prompt was removed to improve the consistency of word length and usage between items.

For the remaining words in the questions, we changed all of the characters to lower case, removed all punctuation marks, numbers, and non-unicode characters. We then stemmed the remaining words to obtain a list of root words. From this list of root words, we removed all words with fewer than three letters. Because we were unsure of the relationship between the words and the labels, we did not create a list of stopwords for removal.

3. TECHNIQUES

We tested four combinations (in no particular order) of word weighting and question labeling algorithms, as shown in Figure 2, to identify the techniques with the highest reliability for our automated learning outcome labeler.

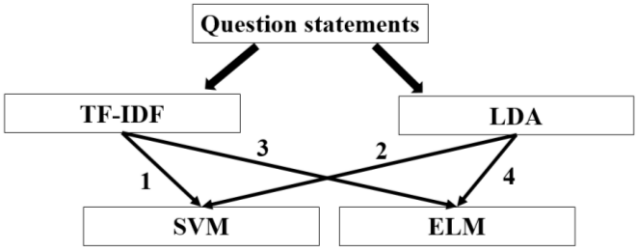


Figure 2: Four combinations of algorithms

Every word in each question prompt was assigned a weightage value based on either term frequency-inverse document frequency (TF-IDF) or latent Dirichlet allocation (LDA). Subsequently, the vector values for each question were passed through either support vector machine (SVM) or extreme learning machine (ELM) to assign a label. All algorithms were implemented in R Studio.

3.1 Term frequency-inverse document frequency

Term frequency-inverse document frequency (TF-IDF) is a technique for finding the relative frequency of words in a given document, and comparing those frequencies with the inverse of how often each of those words appear in the complete document corpus. The resulting ratio can be used to signify the relevance of each unique word within a single document.

We implemented a modified version of TF-IDF that used individual questions as the source of the analysis instead of complete documents. This focused the model on finding the relevance of each word within each single question. By converting each question into a vector of weightages based on word frequencies, the machine learning algorithms were then used to label the questions. The modified TF-IDF model can be described by

$$TF - IDF(w_i, q_k) = \#(w_i, q_k) \times \log \frac{TR}{\#TR(w_i)} \quad (1)$$

where w_i refers to a particular word i, q_k refers to a particular question k, $\#(w_i,q_k)$ refers to number of times w_i occurs in q_k, TR refers to total number of questions and #TR(w_i) refers to question frequency, or the number of questions in which w_i occurs [20].

In the case where the term frequency (TF) count is biased towards longer questions, the TF count is normalized as

$$TF_{i,k} = \frac{n_{i,k}}{\sum_j n_{j,k}} \quad (2)$$

where $n_{i,k}$ refers to the number of times w_i occurs in q_k, the denominator term (size of each question) refers to the sum of the number of times each word appears in q_k [25].

For our work, the pre-processing procedures registered a total of 465 unique stemmed words in our compilation of 120 training questions and 30 testing questions. This led to each question being represented as a vector of 1 row and 465 columns arranged in alphabetical order by stemmed word. When a word is present in a question, the normalized weight of that word is assigned to that question's vector element. If a word is not present in the question, the weight is zero.

After determining the unique word weightage vectors for all 150 questions, the entire matrix is sorted such that for each question, the weightages are arranged in ascending order. The top ten weightages are chosen for each question. The 10 weightages may correspond to different words in each question, but their combinations remain question-specific and give a numerical representation of each question statement. This new vector of 10 columns per question serves as the input to the machine learning algorithms.

As an example, we will use the pre-processed question prompt:

for signal which begin when the one side unilateral ztransform given

Table 2 below shows the weightages assigned to the above example after the application of the TF-IDF technique. The weightages are then arranged in ascending order and the top 10 values are taken.

Table 2 - TF-IDF weightage arrangement

Word (alphabetical order)	**Weightage**
begin	0.392
for	0.140
given	0.140
one	0.222
side	0.356
signal	0.116
the	0.007
unilateral	0.392
when	0.279
which	0.230
ztransform	0.216

3.2 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) is a probabilistic technique for topic modeling based on the Bayesian model. The essential idea of LDA is that each document consists of a mixture of topics, with the continuous-valued mixture properties distributed in a Dirichlet random variable, a continuous multivariate probability distribution.

Again, in the context of our work, we applied LDA to questions in the dataset by substituting the original notion of documents in the LDA algorithm with questions in our modified model. Therefore, the modified model attempted to find *k* number of topics (*k* is a user-defined parameter to determine the desired number of topics, or dimensionality of the Dirichlet distribution) for a given set of question statements based on the choice and usage of words in each question. The joint distribution of a topic mixture, a set of topics and a set of words can be represented by

$$p(\theta, t, w|\alpha, \beta) = p(\theta|\alpha) \prod_{i=1}^{M} p(t_i|\theta) p(w_i|t_i, \beta) \quad (3)$$

where parameter α is a *k*-vector with components more than zero, parameter β refers to the matrix of word probabilities, θ refers to a *k*-dimensional Dirichlet random variable, t_i refers to a topic, w_i refers to a word [26].

Figure 3 shows a graphical model representation of LDA. The bigger circle refers to questions while the smaller circle refers to the repeated choice of topics and words within each question.

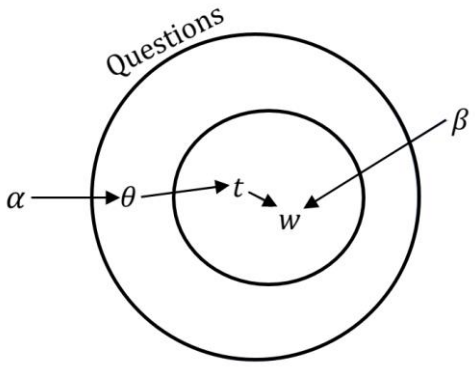


Figure 3: Graphical model representation of LDA

Since LDA involves topic modeling, an appropriate *k* value chosen for our work was ten. This allowed a standard comparison between LDA and the top ten weightages from the TF-IDF method. The generated unique topics (based on the stemmed words) are shown in Table 3.

Table 3 - Topic names generated by LDA

Topic number	Stemmed topic name
1	differ
2	discrete
3	impulse
4	signal
5	filter
6	apply
7	dft
8	output
9	sample
10	system

Out of the entire set of stemmed words detected, ten words have been identified as topic names. Hence, LDA automatically associates the remaining words the above-mentioned ten topics. Based on the words that appear in each question, LDA displays the number of topics per question. Based on the topic assignments, the topic weightages for each question is generated. For topics not present in a question, a minimal weightage is given to those topics in lieu of a zero value. The value ensures that the topic weightages for a question sum to one. Similar to the TF-IDF output, the new vector of 10 columns per question becomes the input for the machine learning algorithms.

3.3 Extreme learning machine

Extreme learning machine (ELM) is a learning algorithm for single-hidden layer feedforward neural networks (SLFNs). ELM can be used for classification, regression, clustering, compression and feature learning. ELM randomly chooses the hidden nodes and determines the output weights of the neural networks.

The following three-step learning model explains ELM. Given a training set that is labeled (information about the target nodes), hidden node activation function and number of hidden nodes,

Step 1: Randomly assign hidden node parameters

Step 2: Calculate the hidden layer output matrix, **H**

Step 3: Calculate the output weight γ

Given a set of inputs with unknown labels, the objective is to find the target outputs [27]. Once the inter-layer weights have been found, the same weights are used during the testing phase. For a given set of input samples x_k, the target/output is given by t_k. For number of hidden nodes *L* and with a certain activation function *f(x)*, the SLFN is modeled as

$$\sum_{j=1}^{L} \gamma_j f_j(x_k) = \sum_{j=1}^{L} \gamma_j f\left(w_j \cdot x_k + b_j\right) = o_k, k = 1, \dots, L \quad (4)$$

where w_j refers to the weight vector that stores the weights between input and hidden nodes, γ_j refers to the weight vector that stores the weights between the hidden and output nodes, b_j refers to the threshold of the jth hidden nodes. The objective is that o_k and t_k (original target) should have zero difference [23] using possible activation functions that include sigmoid, sine, radial basis and hard-limit.

In our case, the output of the ELM are three continuous values that represent the values assigned to the three learning outcome categories (Remember, Apply and Transfer). To convert the three values into a binary value for comparing the predicted labels with the actual labels, we set the learning outcome category with the highest value to one and the remaining two to zero.

3.4 Support vector machine

Support vector machine (SVM) is a mapping of data samples such that these samples can be distinctly labeled. The concept of SVM is derived from margins and subsequently separating data into groups with large gaps between them. Deriving an optimal hyperplane for identifying linearly separable patterns is the key to SVM. This idea is extended to cases where the patterns are non-linearly separable, by using a kernel function to transform the original data samples to map onto a new space [28]. Possible kernels are: linear, polynomial, radial basis and sigmoid.

For our work, we used the C-support vector classification type. Given a set of inputs and targets, the cost function is given by [29]

$$\min_{p,m,\xi} \frac{1}{2} p^T p + C \sum_{j=1}^{k} \xi_j \quad (5)$$

subject to $y_j\left(p^T \phi\left(v_j\right) + m\right) \geq 1 - \xi_j, \xi_j \geq 0, j = 1, \dots, k$

where C>0 is the regularization parameter, m is a constant, p is the vector of coefficients, ξ_j refers to parameters that handle the inputs, index j refers to labeling the k training cases, v refers to the independent variables, y refers to the class labels, ϕ refers to the kernel used that transforms data from the input to the chosen feature space.

Fundamentally, support vectors are data points that lie close to the decision boundary, which are the hardest to classify. SVM maximizes the margin around the hyperplane that separates these points. The cost function is determined based on the training samples (support vectors). These support vectors are the basic elements of a training set that would change the position of the hyperplane dividing the dataset. SVM becomes an optimization problem for determining the optimal hyperplane.

3.5 Performance metrics

To evaluate the reliability of our four technique combinations with the subject matter expert's labels, we looked at using the F1 measure. Accuracy is the number of correct labels divided by the size of testing data. The F1 measure is a harmonic mean of two other metrics: precision and recall. Precision refers to the correctness of questions that have been selected as a particular category. Recall refers to the correctness of selection of the correct category given all the questions that were correctly classified.

Because minimizing the number of false positives and false negatives was important for accurately assigning new questions to the correct practice sets, we used the F1 measure as the basis for our algorithm comparisons. To explain the F1 measure, we will step through the confusion matrix used to describe the performance of a labeling model on a set of testing data. There are four concepts used to construct the confusion matrix:

True positive (TP) refers to the number of questions that the algorithm correctly identifies as presenting a label.

False positive (FP) refers to the number of questions that the algorithm identifies as presenting a label while the subject matter expert indicates the label was absent.

True negative (TN) refers to the number of questions that the algorithm correctly identifies as having a label absent.

False negative (FN) refers to the number of questions that the algorithm identifies as having a label absent while the subject matter expert indicates the label was present.

The F1 measure is calculated as follows [30]

$$Precision = \frac{TP}{(TP+FP)} \quad (6)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (7)$$

$$F1\ measure = \frac{2 \times precision \times recall}{precision+recall} \quad (8)$$

4. RESULTS AND ANALYSIS

4.1 Insights by subject matter expert

When looking at every question presented to students over the course of a semester, the subject matter expert identified the number of questions corresponding to Remember, Apply and Transfer as shown in Table 4. Just by labeling the course questions, the subject matter expert realized how misaligned the course's learning outcomes were with its assessment practices. A large emphasis on Apply questions was expected, but the dearth of Transfer questions was surprising. Of those 23 Transfer items, most were presented during the final exam.

Table 4 - Frequency of questions aligned to learning outcomes

Learning outcome	Frequency (number of questions)
Remember	62
Apply	131
Transfer	23

One of the stated learning outcomes of the course was to prepare students to flexibly transfer course content to novel problems and new situations. However, waiting until the final exam to present students with such opportunities denied them actionable feedback during the semester. In response to the pre-processing labeling efforts, the subject matter expert then added 42 new transfer questions throughout the course for the next semester.

4.2 Model reliability with subject matter expert

The objective of this implementation is to evaluate whether the trained model is able to predict the type of question (Remember, Apply or Transfer). Based on the trained model using questions from the undergraduate course, the testing questions from textbooks and online sources were passed through our model to determine the level of reliability of labeling new questions that were not generated by the subject matter expert. In our intended use case, the testing dataset would not need to be manually labeled. However, to determine the level of reliability of our labeling algorithms, the subject matter expert's manual labels served as a ground truth for the F1 measure calculations.

4.2.1 Parameter selection

We first determined the best set of parameters based on 10-fold cross validation of the training dataset. As there were 120 questions, 90% of the questions (108 questions) were used for training and 10% of the questions (12 questions) were used as a validation set. This process was done 10 times using 10 different bundles of the 120 questions. The best set of parameters were chosen based on a grid search for both ELM and SVM.

The parameters that were varied for ELM were:

1. Number of hidden nodes
2. Activation function (sigmoid / radial basis / hard-limit)

The parameters yielding the best results corresponded to 72 hidden nodes using hard-limit activation function.

The parameters that were varied for SVM were:

1. Kernel (sigmoid / radial basis)
2. Cost value
3. Gamma value

The parameters yielding the best results corresponded to sigmoid kernel, cost value = 1, gamma value = 0.26

4.2.2 Comparing four combinations

With respect to the F1 measure, calculations were done separately for the three labels. The mean of those calculations was then used as the algorithm's overall performance measure. With respect to ELM, the calculation was repeated 10 times because the initialization weights are randomly assigned in each iteration. The mean value of the F1 measure was taken.

Table 5 below shows the F1 measure values (for each individual class and overall F1 mean) for the four combinations. "R" refers to Remember, "A" refers to Apply, "T" refers to Transfer and "s.d." refers to standard deviation.

Table 5 - F1 measure values for four combinations

Combination	R	A	T	Mean	s.d.
1. TF-IDF with SVM	0.870	0.737	0.667	0.758	0.084
2. LDA with SVM	0.400	0.593	0.556	0.516	0.084
3. TF-IDF with ELM	0.926	0.815	0.840	0.860	0.048
4. LDA with ELM	0.467	0.520	0.647	0.545	0.076

TF-IDF with ELM achieved the highest mean F1 measure value and the lowest standard deviation – indicating that it was the most reliable combination. It can be seen that the Remember label yields the highest F1 values out of the three labels in Combination 3. In general, Remember-labeled questions are short, resulting in about four to five zero values in the TF-IDF vector of 10 columns that is passed as an input into the ELM. Hence, the algorithm identifies Remember-labeled questions very accurately due to their size.

The result of high reliability in using ELM is as expected because it has already been demonstrated that ELM outperforms SVM when comparing in terms of standard deviation of training and testing root-mean-square values, time taken, network complexity, as well as performance comparison in real medical diagnosis application [23]. On the other hand, although LDA has been shown to achieve higher performance as it groups words together in terms of topics instead of looking at combinations of individual words which may not link together, in the context of our work, TF-IDF outperforms LDA instead. This is because for LDA, the goal is to correctly assign each document (or question) to a class label in a reduced dimensional space [31]. However, in our corpus of questions, there are several technical terms involved, without any prior labeling of topics. Hence, LDA is not appropriate for our analysis.

5. CONCLUSIONS

Based on the comparison of our four algorithms, our most reliable model (TF-IDF with ELM) is able to accurately label new course questions for the undergraduate electrical and electronic engineering course with 0.86 reliability in terms of F1 measure. Any novice instructor who takes over this course in the future or teaching assistants tasked with refreshing the course assignments would be able to extract new questions from any external source and pass them to the algorithm to automatically label the questions as the original course coordinator would. This allows members of the course design team without a strong background in learning to make curriculum decisions regarding the alignment of the course's learning outcomes.

As discussed earlier, outcome-based learning environments facilitate transforming the model of instruction from instructor-centric and lecture-based to being more learner focused filled with a variety of activities and learning pathways. However, in learner-centered environments, assessment is still the key driver, and often the key inhibitor of learning [3]. If the assessments require shallow understanding, then learners calibrate their efforts to achieve this low bar. When assessments require deep understanding or great proficiency, learners are likely to put in more effortful practice.

In line with this assessment philosophy, our TF-IDF with ELM model is theoretically capable of matching any learning activity to any set of learning outcomes as long as the course designers or subject matter experts provide enough examples that are explicitly aligned to the intended learning outcomes when training the model. For the convenience of the subject matter expert in our context, we used a reduced version of Bloom's Taxonomy in this study. However, the final algorithm is capable of using the full Bloom's model, a different model, or a custom set of learning outcomes as its labeling framework.

Hence, with the high reliability of the prediction algorithm presented in our work, our process for calibrating the algorithm can be used in any academic or industrial setting to provide the right set of formative assessment opportunities to students (enhancing subject knowledge) or employees (professional development). Once the learning outcomes of activities are labeled reliably, it is then easier to think about how to engage learners in deliberate practice to reach those outcomes and develop their expertise. Once opportunities for deliberate practice that align to the course learning outcomes are implemented into a course, it becomes easier to think about how to align the feedback regarding those opportunities to support the development of domain expertise.

This work provides a first step at being able to regularly introduce learning activities that promote the development of adaptive expertise into a course by matching external sources of activities with the course's learning outcomes. Deliberate practice requires repetition that varies in ways that highlight the structural elements of a domain. Having a way to incorporate new sources of questions and problems into a course that align with the course's goals provides learners more opportunities for internalizing when to apply their domain specific skills and knowledge. Finally, our algorithm is potentially useful for designing courses to reach non-content-based learning outcomes, making policies that support constructive alignment, and evaluating course assessment of learning plans.

6. FUTURE WORK

Building off of our machine learning labeling work, we would like to explore constructing a new version of LDA that can be tailor-made to label questions. There are situations in which weightages given to words are the same, with different words representing those weightages. Similarly, the same words can have different weightages. We are keen to continue working on features based on word arrangement, word context and word order that affect weightage assignments. In addition, ELM can be enhanced by using kernels.

From the learning aspect, we would like to extend our question label categories to all six outcomes described in Bloom's Taxonomy and expand the model to label outcomes based on the types of sentences used in forum conversations and other collaborative learning activities. Eventually, we aim to determine the proficiency level of learners so we can put learning supports in place to guide their learning journeys. Ultimately, we wish to provide learners with learning activities and opportunities for deliberate practice embedded with actionable feedback to develop their adaptive expertise.

7. ACKNOWLEDGMENTS

This work was conducted within the Delta-NTU Corporate Lab for Cyber-Physical Systems with funding support from Delta Electronics Inc and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

8. REFERENCES

[1] Krathwohl, D.R. 2002. A Revision of Bloom's Taxonomy: An Overview. *Theory into Practice*. 41, 4 (2002), 212-218. DOI= http://dx.doi.org/10.1207/s15430421tip4104_2

[2] Biggs, J. 1996. Enhancing teaching through constructive alignment. *Higher Education*. 32, 3 (1996), 347-364. DOI= http://dx.doi.org/10.1007/BF00138871

[3] Boud, D. 2010. Sustainable Assessment: Rethinking assessment for the learning society. *Studies in Continuing Education*. 22, 2 (2010), 151-167. DOI= http://dx.doi.org/10.1080/713695728

[4] Boud, D. and Falchikov, N. 2006. Aligning assessment with long-term learning. *Assessment & Evaluation in Higher Education*. 31, 4 (2006), 399-413. DOI= http://dx.doi.org/10.1080/02602930600679050

[5] Miller, G. E. 1990. The Assessment of Clinical Skills/Competence/Performance. *Academic Medicine*. 65, 9 (1990), S63-S67. DOI= http://dx.doi.org/10.1097/00001888-199009000-00045

[6] Wass, V. et al. 2001. Assessment of clinical competence. *The Lancet*. 357, 9260 (2001), 945-949. DOI= http://dx.doi.org/10.1016/S0140-6736(00)04221-5

[7] Hmelo-Silver, C.E. 2004. Problem-based learning: What and how do students learn? *Educational Psychology Review*. 16, 3 (2004). 235-266. DOI= http://dx.doi.org/10.1023/B:EDPR.0000034022.16470.f3

[8] Biggs, J. B. and Collis, K.F. 2014. *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcomes).* Academic Press.

[9] Crowe, A. et al. 2008. Biology in bloom: implementing Bloom's taxonomy to enhance student learning in biology. *CBE-Life Sciences Education.* 7, 4 (2008), 368-381. DOI= http://dx.doi.org/10.1187/cbe.08-05-0024

[10] Ericsson, K.A. et al. 1993. The Role of Deliberate Practice in the Acquisition of Expert Performance. *Psychological Review.* 100, 3 (1993), 363-406. DOI= http://dx.doi.org/10.1037/0033-295X.100.3.363

[11] Duckworth, A. L. et al. 2007. Grit: perseverance and passion for long-term goals. *Journal of personality and social psychology*. 92, 6 (2007), 1087. DOI= http://dx.doi.org/10.1037/0022-3514.92.6.1087

[12] Schwartz D. L. et al. 2005. Efficiency and innovation in transfer. *Transfer of learning from a Modern Multidisciplinary Perspective*. Information Age Publishing. 1-51.

[13] Chi, M. T. 2006. Two approaches to the study of experts' characteristics. *The Cambridge Handbook of expertise and expert* performance. Cambridge University Press. 21-30.

[14] Black, P. and William, D. 1998. Assessment and Classroom Learning. *Assessment in Education Principles Policy and Practice.* 5, 1 (1998), 7-74. DOI= http://dx.doi.org/10.1080/0969595980050102

[15] Ritter, S. et al. 2007. Cognitive Tutor: Applied research in mathematics education. *Psychonomic bulletin & review.* 14, 2 (2007), 249-255. DOI= http://dx.doi.org/10.3758/BF03194060

[16] Fei, T. et al. 2003. Question Classification for E-learning by Artificial Neural Network. In *Proceedings of the 2003 Joint Fourth International Conference on Information, Communications and Signal Processing and the Fourth Pacific Rim Conference on Multimedia* (Singapore, 2003), 1-5. DOI= http://dx.doi.org/10.1109/ICICS.2003.1292768

[17] Cheng, S. C. et al. 2005. Automatic Leveling System for E-Learning Examination Pool Using Entropy-Based Decision Tree. In *Advances in Web-Based Learning – ICWL 2005* (Hong Kong, 2005), 273-278. DOI= http://dx.doi.org/10.1007/11528043_27

[18] Jayakodi, K. et al. 2015. An Automatic Classifier for Exam Questions in Engineering: A Process for Bloom's Taxonomy. In *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* (Zhuhai, China, 2015). DOI= https://dx.doi.org/10.1109/TALE.2015.7386043

[19] Haris, S. S. and Omar, N. 2015. Bloom's taxonomy question categorization using rules and N-gram approach. *Journal of Theoretical and Applied Information Technology.* 76, 3 (2015), 401-407.

[20] Yahya, A. A. et al. 2013. Analyzing the cognitive level of classroom questions using machine learning techniques. In *The 9th International Conference on Cognitive Science* (Kuching, Sarawak, Malaysia, 2013). 587-595. DOI= http://dx.doi.org/10.1016/j.sbspro.2013.10.277

[21] Abduljabbar, D. A. and Omar, N. 2015. Exam questions classification based on Bloom's taxonomy cognitive level using classifiers combination. *Journal of Theoretical and Applied Information Technology.* 78, 3 (2015), 447-455.

[22] Sangodiah, A. et al. 2014. A Review in Feature Extraction Approach in Question Classification Using Support Vector Machine. In *2014 IEEE International Conference on Control System, Computing and Engineering* (Penang, Malaysia, 2014), 536-541. DOI= http://dx.doi.org/10.1109/ICCSCE.2014.7072776

[23] Huang, G. B. et al. 2006. Extreme learning machine: Theory and applications. *Neurocomputing.* 70, 1-3 (2006), 489-501. DOI= http://dx.doi.org/10.1016/j.neucom.2005.12.126

[24] Trinity University Course Assessment and Outcomes: 2016 *https://inside.trinity.edu/collaborative/collaborative-grants/course-redesign-stipends/course-assessment-and-outcomes*. Accessed: 2017-02-24.

[25] Bernardi, R. *Term Frequency and Inverted Document Frequency.* University of Trento, Trentino.

[26] Blei, D. M. et al. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research.* 3 (2003), 993-1022.

[27] Huang, G. B. 2015. What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt's Dream and John von Neumann's Puzzle. *Cognitive Computation.* 7, 3 (2015), 263-278. DOI= http://dx.doi.org/10.1007/s12559-015-9333-0

[28] Weston, J. *Support Vector Machine (and Statistical Learning Theory).* NEC Labs America, Princeton.

[29] Chang, C. C. and Lin, C. J. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST).* 2, 3 (2011), 1-39. DOI= http://dx.doi.org/10.1145/1961189.1961199

[30] Santra, A. K. and Christy, C. J. 2012. Genetic Algorithm and Confusion Matrix for Document Clustering. *IJCSI International Journal of Computer Science Issues.* 9, 1 (2012), 322-328.

[31] Hu, D. J. 2009. *Latent Dirichlet Allocation for Text, Images, and Music*.

Behavior-Based Latent Variable Model for Learner Engagement

Andrew S. Lan[1], Christopher G. Brinton[2], Tsung-Yen Yang[3], Mung Chiang[1]
[1]Princeton University, [2]Zoomi Inc., [3]National Chiao Tung University
andrew.lan@princeton.edu, christopher.brinton@zoomiinc.com, tsungyenyang.eecs02@nctu.edu.tw, chiangm@princeton.edu

ABSTRACT
We propose a new model for learning that relates video-watching behavior and engagement to quiz performance. In our model, a learner's knowledge gain from watching a lecture video is treated as proportional to their latent engagement level, and the learner's engagement is in turn dictated by a set of behavioral features we propose that quantify the learner's interaction with the lecture video. A learner's latent concept knowledge is assumed to dictate their observed performance on in-video quiz questions. One of the advantages of our method for determining engagement is that it can be done entirely within standard online learning platforms, serving as a more universal and less invasive alternative to existing measures of engagement that require the use of external devices. We evaluate our method on a real-world massive open online course (MOOC) dataset, from which we find that it achieves high quality in terms of predicting unobserved first-attempt quiz responses, outperforming two state-of-the-art baseline algorithms on all metrics and dataset partitions tested. We also find that our model enables the identification of key behavioral features (e.g., larger numbers of pauses and rewinds, and smaller numbers of fast forwards) that are correlated with higher learner engagement.

Keywords
Behavioral data, engagement, latent variable model, learning analytics, MOOC, performance prediction

1. INTRODUCTION
The recent and rapid development of online learning platforms, coupled with advancements in machine learning, has created an opportunity to revamp the traditional "one-size-fits-all" approach to education. This opportunity is facilitated by the ability of many learning platforms, such as massive open online course (MOOC) platforms, to collect several different types of data on learners, including their assessment responses as well as their learning behavior [9]. The focus of this work is on using different forms of data to model the learning process, which can lead to effective learning analytics and potentially improve learning efficacy.

1.1 Behavior-based learning analytics
Current approaches to learning analytics are focused mainly on providing feedback to learners about their knowledge states – or the level to which they have mastered given concepts/topics/knowledge components – through analysis of their responses to assessment questions [10, 24]. There are other cognitive (e.g., engagement [17, 31], confusion [37], and emotion [11]) as well as non-cognitive (e.g., fatigue, motivation, and level of financial support [14]) factors beyond assessment performance that are crucial to the learning process as well. Accounting for them thus has the potential to yield more effective learning analytics and feedback.

To date, it has been difficult to measure these factors of the learning process. Contemporary online learning platforms, however, have the capability to collect *behavioral data* that can provide some indicators of them. This data commonly includes learners' usage patterns of different types of learning resources [12, 15], their interactions with others via social learning networks [7, 28], their clickstream and keystroke activity logs [2, 8, 30], and sometimes other metadata including facial expressions [35] and gaze location [6].

Recent research has attempted to use behavioral data to augment learning analytics. [5] proposed a latent response model to classify whether a learner is gaming an intelligent tutoring system, for example. Several of these works have sought to demonstrate the relationship between behavior and performance of learners in different scenarios. In the context of MOOCs, [22] concluded that working on more assignments lead to better knowledge transfer than only watching videos, [12] extracted probabilistic use cases of different types of learning resources and showed they are predictive of certification, [32] used discussion forum activity and topic analysis to predict test performance, and [26] discovered that submission activities can be used to predict final exam scores. In other educational domains, [2] discovered that learner keystroke activity in essay-writing sessions is indicative of essay quality, [29] identified behavior as one of the factors predicting math test achievement, and [25] found that behavior is predictive of whether learners can provide elegant solutions to mathematical questions.

In this work, we are interested in how behavioral data can be used to model a learner's *engagement*.

1.2 Learner engagement
Monitoring and fostering engagement is crucial to education, yet defining it concretely remains elusive. Research has sought to identify factors in online learning that may drive engagement; for example, [17] showed that certain production styles of lecture videos promote it. [20] defined disengagement as dropping out in the middle of a video and studied the relationship between disengagement and video content, while [31] considered the relationship between engagement and the

semantic features of mathematical questions that learners respond to. [33] studied the relationship between learners' self-reported engagement levels in a learning session and their facial expressions immediately following in-session quizzes, and [34] considered how engagement is related to linguistic features of discussion forum posts.

There are many types of engagement [3], with the type of interest depending on the specific learning scenario. Several approaches have been proposed for measuring and quantifying different types. These approaches can be roughly divided into two categories: device-based and activity-based. Device-based approaches measure learner engagement using devices external to the learning platform, such as cameras to record facial expressions [35], eye-tracking devices to detect mind wandering while reading text documents [6], and pupil dilation measurements, which are claimed to be highly correlated with engagement [16]. Activity-based approaches, on the other hand, measure engagement using heuristic features constructed from learners' activity logs; prior work includes using replies/upvote counts and topic analysis of discussions [28], and manually defining different engagement levels based on activity types found in MOOCs [4, 21].

Both of these types have their drawbacks. Device-based approaches are far from universal in standard learning platforms because they require integration with external devices. They are also naturally invasive and carry potential privacy risks. Activity-based approaches, on the other hand, are not built on the same granularity of data, and tend to be defined from heuristics that have no guarantee of correlating with learning outcomes. It is therefore desirable to develop a statistically principled, activity-based approach to inferring a learner's engagement.

1.3 Our approach and contributions

In this paper, we propose a probabilistic model for inferring a learner's engagement level by treating it as a latent variable that drives the learner's performance and is in turn driven by the learner's behavior. We apply our framework to a real-world MOOC dataset consisting of clickstream actions generated as learners watch lecture videos, and question responses from learners answering in-video quiz questions.

We first formalize a method for quantifying a learner's behavior while watching a video as a set of nine *behavioral features* that summarize the clickstream data generated (Section 2). These features are intuitive quantities such as the fraction of video played, the number of pauses made, and the average playback rate, some of which have been associated with performance previously [8]. Then, we present our statistical model of learning (Section 3) as two main components: a *learning model* and a *response model*. The learning model treats a learner's gain in concept knowledge as proportional to their latent engagement level while watching a lecture video. Concept knowledge is treated as multidimensional, on a set of latent concepts underlying the course, and videos are associated with varying levels to different concepts. The response model treats a learner's performance on in-video quiz questions, in turn, as proportional to their knowledge on the concepts that this particular question relates to.

By defining engagement to correlate directly with performance, we are able to learn which behavioral features lead to high engagement through a single model. This differs from prior works that first define heuristic notions of engagement and subsequently correlate engagement with performance, in separate procedures. Moreover, our formulation of latent engagement can be made from entirely within standard learning platforms, serving as a more universally applicable and less invasive alternative to device-based approaches.

Finally, we evaluate two different aspects of our model (Section 4): its ability to predict unobserved, first-attempt quiz question responses, and its ability to provide meaningful analytics on engagement. We find that our model predicts with high quality, achieving AUCs of up to 0.76, and outperforming two state-of-the-art baselines on all metrics and dataset partitions tested. One of the partitions tested corresponds to the beginning of the course, underscoring the ability of our model to provide early detection of struggling or advanced students. In terms of analytics, we find that our model enables us to identify behavioral features (e.g., large numbers of pauses and rewinds, and small numbers of fast forwards) that indicate high learner engagement, and to track learners' engagement patterns throughout the course. More generally, these findings can enable an online learning platform to detect learner disengagement and perform appropriate interventions in a fully automated manner.

2. BEHAVIORAL DATA

In this section, we start by detailing the setup of lecture videos and quizzes in MOOCs. We then specify video-watching clickstream data and our method for summarizing it into behavioral features.

2.1 Course setup and data capture

We are interested in modeling learner engagement while watching lecture videos to predict their performance on in-video quiz questions. For this purpose, we can view an instructor's course delivery as the sequence of videos that learners will watch interspersed with the quiz questions they will answer. Let $Q = (q_1, q_2, \ldots)$ be the sequence of questions asked through the course. A video could have any number of questions generally, including none; to enforce a 1:1 correspondence between video content and questions, we will consider the "video" for question q_n to be all video content that appears between q_{n-1} and q_n. Based on this, we will explain the formats of video-watching and quiz response data we work with in this section.

Our dataset. The dataset we will use is from the fall 2012 offering of the course *Networks: Friends, Money, and Bytes* (FMB) on Coursera [1]. This course has 92 videos distributed among 20 lectures, and exactly one question per video.

2.1.1 Video-watching clickstreams

When a learner watches a video on a MOOC, their behavior is typically recorded as a sequence of clickstream actions. In particular, each time a learner makes an action – `play`, `pause`, `seek`, `ratechange`, `open`, or `close` – on the video player, a clickstream event is generated. Formally, the ith event created for the course will be in the format

$$E_i =< u_i, v_i, e_i, p_i', p_i, x_i, s_i, r_i >$$

Here, u_i and v_i are the IDs of the specific learner (user) and video, respectively, and e_i is the type of action that u_i made on v_i. p_i is the position of the video player (in seconds) immediately after e_i is made, p'_i is the position immediately before,[1] x_i is the UNIX timestamp (in seconds) at which e_i was fired, s_i is the binary state of the video player – either `playing` or `paused` – once this action is made, and r_i is the playback rate of the video player once this action is made. Our FMB dataset has 314,632 learner-generated clickstreams from 3,976 learners.[2]

The set $E_{u,v} = \{E_i | u_i = u, v_i = v\}$ of clickstreams for learner u recorded on video v can be used to reconstruct the behavior u exhibits on v. In Section 2.2 we will explain the features computed from $E_{u,v}$ to summarize this behavior.

2.1.2 *Quiz responses*

When a learner submits a response to an in-video quiz question, an event is generated in the format

$$A_m =< u_m, v_m, x_m, a_m, y_m >$$

Again, u_m and v_m are the learner and video IDs (*i.e.,* the quiz corresponding to the video). x_m is the UNIX timestamp of the submission, a_m is the specific response, and y_m is the number of points awarded for the response. The questions in our dataset are multiple choice with a single response, so y_m is binary-valued.

In this work, we are interested in whether quiz responses were correct on first attempt (CFA) or not. As a result, with $A_{u,v} = \{A_m | u_m = u, v_m = v\}$, we consider the event $A'_{u,v}$ in this set with the earliest timestamp $x'_{u,v}$. We also only consider the set of clickstreams $E'_{u,v} \subseteq E_{u,v}$ that occur before $x'_{u,v}$, as the ones after would be anti-causal to CFA.

2.2 Behavioral features and CFA score

With the data $E'_{u,v}$ and $A'_{u,v}$, we construct two sets of information for each learner u on each video v, *i.e.,* each learner-video pair. First is a set of nine behavioral features that summarize u's video-watching behavior on v [8]:

(1) Fraction spent. The fraction of time the learner spent on the video, relative to the playback length of the video. Formally, this quantity is $e_{u,v}/l_v$, where

$$e_{u,v} = \sum_{i \in \mathcal{S}} \min(x_{i+1} - x_i, l_v)$$

is the elapsed time on v obtained by finding the total UNIX time for u on v, and l_v is the length of the video (in seconds). Here, $\mathcal{S} = \{i \in A'_{u,v} : a_{i+1} \neq \texttt{open}\}$. l_v is included as an upper bound for excessively long intervals of time.

(2) Fraction completed. The fraction of the video that the learner completed, between 0 (none) and 1 (all). Formally, it is $c_{u,v}/l_v$, where $c_{u,v}$ is the number of unique 1 second segments of the video that the learner visited.

[1] p_i and p'_i will only differ when i is a skip event.

[2] This number excludes invalid `stall`, `null`, and `error` events, as well as `open` and `close` events which are generated automatically.

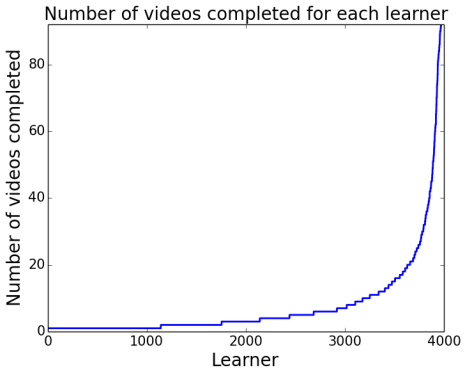


Figure 1: Distribution of the number of videos that each each learner completed in FMB. More than 85% of learners completed less than 20 videos.

(3) Fraction played. The fraction of the video that the learner played relative to the length. Formally, it is calculated as $g_{u,v}/l_v$, where

$$g_{u,v} = \sum_{i \in \mathcal{S}} \min(p'_{i+1} - p_i, l_v)$$

is the total length of video that was played (while in the playing state). Here, $\mathcal{S} = \{i \in A'_{u,v} : a_{i+1} \neq \texttt{open} \ \wedge \ s_i = \texttt{playing}\}$.

(4) Fraction paused. The fraction of time the learner stayed paused on the video relative to the length. It is calculated as $h_{u,v}/l_v$, where

$$h_{u,v} = \sum_{i \in \mathcal{S}} \min(t_{i+1} - t_i, l_v)$$

is the total time the learner stayed in the `paused` state on this video. Here, $\mathcal{S} = \{i \in A'_{u,v} : a_{i+1} \neq \texttt{open} \ \wedge \ s_i = \texttt{paused}\}$.

(5) Number of pauses. The number of times the learner paused the video, or

$$\sum_{i \in A'_{u,v}} \mathbb{1}\{a_i = \texttt{pause}\}$$

where $\mathbb{1}\{\}$ is the indicator function.

(6) Number of rewinds. The number of times the learner skipped backwards in the video, or

$$\sum_{i \in A'_{u,v}} \mathbb{1}\{a_i = \texttt{skip} \ \wedge \ p'_i < p_i\}$$

(7) Number of fast forwards. The number of times the learner skipped forward in the video, *i.e.,* with $p'_i > p_i$ in the previous equation.

(8) Average playback rate. The time-average of the learner's playback rate on the video. Formally, it is calculated as

$$\bar{r}_{u,v} = \frac{\sum_{i \in \mathcal{S}} r_i \cdot \min(x_{i+1} - x_i, l_v)}{\sum_{i \in \mathcal{S}} \min(x_{i+1} - x_i, l_v)}$$

where $\mathcal{S} = \{i \in A'_{u,v} : a_{i+1} \neq \texttt{open} \ \wedge \ s_i = \texttt{playing}\}$.

(9) Standard deviation of playback rate. The standard deviation of the learner's playback rate. It is calculated as

$$\sqrt{\frac{\sum_{i\in\mathcal{S}}(r_i-\bar{r}_{u,v})^2\cdot\min(x_{i+1}-x_i,l_v)}{\sum_{i\in\mathcal{S}}\min(x_{i+1}-x_i,l_v)}}$$

with the same $\mathcal{S}$ as the average playback rate.

The second piece of information for each learner-video pair is u's CFA score $y_{u,v}\in\{0,1\}$ on the quiz question for v.

2.3 Dataset subsets

We will consider different groups of learner-video pairs when evaluating our model in Section 4. Our motivation for doing so is the heterogeneity of learner motivation and high dropoff rates in MOOCs [9]: many will quit the course after watching just a few lectures. Modeling in a small subset of data, particularly those at the beginning of the course, is desirable because it can lead to "early detection" of those who may drop out [8].

Figure 1 shows the dropoff for our dataset in terms of the number of videos each learner completed: more than 85% of learners completed just 20% of the course. "Completed" is defined here as having watched some of the video and responded to the corresponding question. Let T_u be the number of videos learner u completed and $\gamma(v)$ be the index of video v in the course, we define $\Omega^{u_0,v_0}=\{(u,v):T_u\geq u_0\ \wedge\ \gamma(v)\leq v_0\}$ to be the subset of learner-video pairs such that u completed at least u_0 videos and v is within the first v_0 videos. The full dataset is $\Omega^{1,92}$, and we will also consider $\Omega^{20,92}$ as the subset of 346 active learners over the full course and $\Omega^{1,20}$ as the subset of all learners over the first two weeks[3] in our evaluation.

3. STATISTICAL MODEL OF LEARNING WITH LATENT ENGAGEMENT

In this section, we propose our statistical model. Let U denote the number of learners (indexed by u) and V the number of videos (indexed by v). Further, we use T_u to denote the number of time instances registered by learner u (indexed by t); we take a time instance to be a learner completing a video, i.e., watching a video and answering the corresponding quiz question. For simplicity, we use a discrete notion of time, i.e., each learner-video pair will correspond to one time instance for one learner.

Our model considers learners' responses to quiz questions as measurements of their underlying knowledge on a set of concepts; let K denote the number of such concepts. Further, our model considers the action of watching lecture videos as part of learning that changes learners' latent knowledge states over time. These different aspects of the model are visualized in Figure 2: there are two main components, a response model and a learning model.

3.1 Response Model

Our statistical model of learner responses is given by

$$p(y_u^{(t)}=1|\mathbf{c}_u^{(t)})=\sigma(\mathbf{w}_{v(u,t)}^T\mathbf{c}_u^{(t)}-\mu_{v(u,t)}+a_u),\qquad(1)$$

[3]In FMB, the first two weeks of lectures is the first 20 videos.

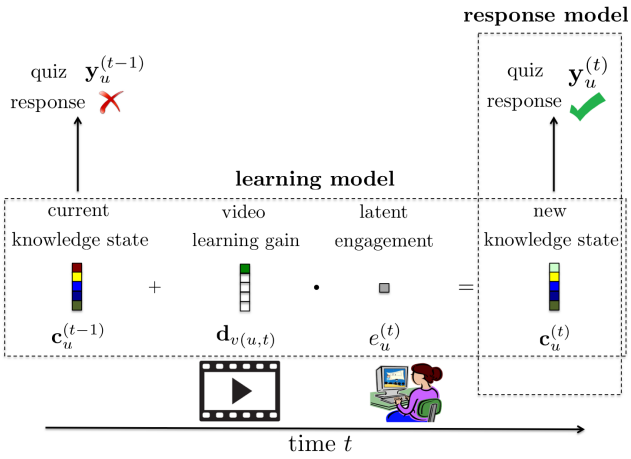


Figure 2: Our proposed statistical model of learning consists of two main parts, a response model and a learning model.

where $v(u,t):\Omega\subseteq\{1,\ldots,U\}\times\{1,\ldots,\max_u T_u\}\rightarrow\{1,\ldots,V\}$ denotes a mapping from a learner index-time index pair to the index of the video v that u was watching at t. $y_u^{(t)}\in\{0,1\}$ is the binary-valued CFA score of learner u on the quiz question corresponding to the video they watch at time t, with 1 denoting a correct response (CFA) and 0 denoting an incorrect response (non-CFA).

The variable $\mathbf{w}_v\in\mathbb{R}_+^K$ denotes the non-negative, K-dimensional quiz question–concept association vector that characterizes how the quiz question corresponding to video v tests learners' knowledge on each concept, and the variable μ_v is a scalar characterizing the intrinsic difficulty of the quiz question. $\mathbf{c}_u^{(t)}$ is the K-dimensional concept knowledge vector of learner u at time t, characterizing the knowledge level of the learner on each concept at the time, and a_u denotes the static, intrinsic ability of learner u. Finally, $\sigma(x)=\frac{1}{1+e^{-x}}$ is the sigmoid function.

We restrict the question–concept association vector $\mathbf{w}_v$ to be non-negative in order to make the parameters interpretable [24]. Under this restriction, the values of concept knowledge vector $\mathbf{c}_u^{(t)}$ can be understood as follows: large, positive values lead to higher chances of answering a question correctly, thus corresponding to high knowledge, while small, negative values lead to lower chances of answering a question correctly, thus corresponding to low knowledge.

3.2 Learning Model

Our model of learning considers transitions in learners' knowledge states as induced by watching lecture videos. It is given by

$$\mathbf{c}_u^{(t)}=\mathbf{c}_u^{(t-1)}+e_u^{(t)}\mathbf{d}_{v(u,t)},\quad t=1,\ldots,T_u,\qquad(2)$$

where the variable $\mathbf{d}_v\in\mathbb{R}_+^K$ denotes the non-negative, K-dimensional learning gain vector for video v; each entry characterizes the degree to which the video improves learners' knowledge level on each concept. The assumption of non-negativity on $\mathbf{d}_v$ implies that videos will not negatively affect learners' knowledge, as in [23]. $\mathbf{c}_u^{(0)}$ is the initial knowledge state of learner u at time $t=0$, i.e., before starting the

	$\Omega^{20,92}$		$\Omega^{1,20}$		$\Omega^{1,92}$	
	ACC	AUC	ACC	AUC	ACC	AUC
Proposed model	**0.7293±0.0070**	**0.7608±0.0094**	**0.7096±0.0057**	**0.7045±0.0066**	**0.7058±0.0054**	**0.7216±0.0054**
SPARFA	0.7209±0.0070	0.7532±0.0098	0.7061±0.0069	0.7020±0.0070	0.6975±0.0048	0.7124±0.0050
BKT	0.7038±0.0084	0.7218±0.0126	0.6825±0.0058	0.6662±0.0065	0.6803±0.0055	0.6830±0.0059

Table 1: Quality comparison of the different algorithms on predicting unobserved quiz question responses. The obtained ACC and AUC metrics on different subsets of the FMB dataset are given. Our proposed model obtains higher quality than the SPARFA and BKT baselines in each case.

course and watching any video.

The scalar latent variable $e_u^{(t)} \in [0, 1]$ in (2) characterizes the *engagement level* that learner u exhibits when watching video $v(u, t)$ at time t. This is in turn modeled as

$$e_u^{(t)} = \sigma(\boldsymbol{\beta}^T \mathbf{f}_u^{(t)}), \tag{3}$$

where $\mathbf{f}_u^{(t)}$ is a 9-dimensional vector of the behavioral features defined in Section 2.2, summarizing learner u's behavior while the video at time t. $\boldsymbol{\beta}$ is the unknown, 9-dimensional parameter vector that characterizes how engagement associates with each behavioral feature.

Taken together, (2) and (3) state that the knowledge gain a learner will experience on a particular concept while watching a particular video is given by

(i) the video's intrinsic association with the concept, modulated by

(ii) the learner's engagement while watching the video, as manifested by their clickstream behavior.

From (2), a learner's (latent) engagement level dictates the fraction of the video's available learning gain they acquire to improve their knowledge on each concept. The response model (1) in turn holds that performance is dictated by a learner's concept knowledge states. In this way, engagement is directly correlated with performance through the concept knowledge states. Note that in this paper, we treat the engagement variable $e_u^{(t)}$ as a scalar; the extension of modeling it as a vector and thus separating engagement by concept is part of our ongoing work.

It is worth mentioning the similarity between our characterization of engagement as a latent variable in the learning model and the input gate variables in long-short term memory (LSTM) neural networks [18]. In LSTM, the change in the latent memory state (loosely corresponding to the latent concept knowledge state vector $\mathbf{c}_u^{(t)}$) is given by the input vector (loosely corresponding to the video learning gain vector $\mathbf{d}_v$) modulated by a set of input gate variables (corresponding to the engagement variable $e_u^{(t)}$).

Parameter inference. Our statistical model of learning and response can be seen as a particular type of recurrent neural network (RNN). Therefore, for parameter inference, we implement a stochastic gradient descent algorithm with standard backpropagation. Given the graded learner responses $y_u^{(t)}$ and behavioral features $\mathbf{f}_u^{(t)}$, our parameter inference algorithm estimates the quiz question–concept association vectors $\mathbf{w}_v$, the quiz question intrinsic difficulties μ_v, the the video learning gain vectors $\mathbf{d}_v$, the learner initial knowledge vectors $\mathbf{c}_u^{(0)}$, the learner abilities a_u, and the engagement–behavioral feature association vector $\boldsymbol{\beta}$. We omit the details of the algorithm for simplicity of exposition.

4. EXPERIMENTS

In this section, we evaluate the proposed latent engagement model on the FMB dataset. We first demonstrate the gain in predictive quality of the proposed model over two baseline algorithms (Section 4.1), and then show how our model can be used to study engagement (Section 4.2).

4.1 Predicting unobserved responses

We evaluate our proposed model's quality by testing its ability to predict unobserved quiz question responses.

Baselines. We compare our model against two well-known, state-of-the-art response prediction algorithms that do not use behavioral data. First is the sparse factor analysis (SPARFA) algorithm [24], which factors the learner-question matrix to extract latent concept knowledge, but does not use a time-varying model of learners' knowledge states. Second is a version of the Bayesian knowledge tracing (BKT) algorithm that tracks learners' time-varying knowledge states, which incorporates a set of guessing and slipping probability parameters for each question, a learning probability parameter for each video, and an initial knowledge level parameter for each learner [13, 27].

4.1.1 Experimental setup and metrics

Regularization. In order to prevent overfitting, we add ℓ_2-norm regularization terms to the overall optimization objective function for every set of variables in both the proposed model and in SPARFA. We use a parameter λ to control the amount of regularization on each variable.

Cross validation. We perform 5-fold cross validation on the full dataset ($\Omega^{1,92}$), and on each subset of the dataset introduced in Section 2.3 ($\Omega^{20,92}$ and $\Omega^{1,20}$). To do so, we randomly partition each learner's quiz question responses into 5 data folds. Leaving out one fold as the test set, we use the remaining four folds as training and validation sets to select the values of the tuning parameters for each algorithm, i.e., by training on three of the folds and validating on the other. We then train every algorithm on all four observed folds using the tuned values of the parameters, and evaluate them on the holdout set. All experiments are repeated for 20 random partitions of the training and test sets.

For the proposed model and for SPARFA, we tune both the

Feature	Coefficient
Fraction spent	0.1941
Fraction completed	0.1443
Fraction played	0.2024
Fraction paused	0.0955
Number of pauses	0.2233
Number of rewinds	0.4338
Number of fast forwards	−0.1551
Average playback rate	0.2797
Standard deviation of playback rate	0.0314

Table 2: Regression coefficient vector $\boldsymbol{\beta}$ learned over the full dataset, associating each clickstream feature to engagement. All but one of the features (number of fast forwards) is positively correlated with engagement.

Figure 3: Plot of the latent engagement level $e_j^{(t)}$ over time for one third of the learners in FMB, showing a diverse set of behaviors across learners.

number of concepts $K \in \{2, 4, 6, 8, 10\}$ and the regularization parameter $\lambda \in \{0.5, 1.0, \ldots, 10.0\}$. Note that for the proposed model, when a question response is left out as part of the test set, only the response is left out of the training set: the algorithm still uses the clickstream data for the corresponding learner-video pair to model engagement.

Metrics. To evaluate the quality of the algorithms, we employ two commonly used binary classification metrics: prediction accuracy (ACC) and area under the receiver operating characteristic curve (AUC) [19]. The ACC metric is simply the fraction of predictions that are made correctly, while the AUC measures the tradeoff between the true and false positive rates of the classifier. Both metrics take values in $[0, 1]$, with larger values indicating higher quality.

4.1.2 *Results and discussion*

Table 1 gives the evaluation results for the three algorithms. The average and standard deviation over the 20 random data partitions are reported for each dataset group and metric.

First of all, the results show that our proposed model consistently achieves higher quality than both baseline algorithms on both metrics. It significantly outperforms BKT in particular (SPARFA also outperforms BKT). This shows the potential of our model to push the envelope on achievable quality in performance prediction research.

Notice that our model achieves its biggest quality improvement on the full dataset, with a 1.3% gain in AUC over SPARFA and a 5.7% gain over BKT. This observation suggests that as more clickstream data is captured and available for modeling – especially as we observe more video-watching behavioral data from learners over a longer period of time (the full dataset $\Omega^{1,92}$ contains clickstream data for up to 12 weeks, while the $\Omega^{1,20}$ subset only contains data for the first 2 weeks) – the proposed model achieves more significant quality enhancements over the baseline algorithms. This is somewhat surprising, since prior work on behavior-based performance prediction [8] has found the largest gains in the presence of fewer learner-video pairs, i.e., before there are many question responses for other algorithms to model on. But our algorithm also benefits from additional question responses, to update its learned relationship between behavior and concept knowledge.

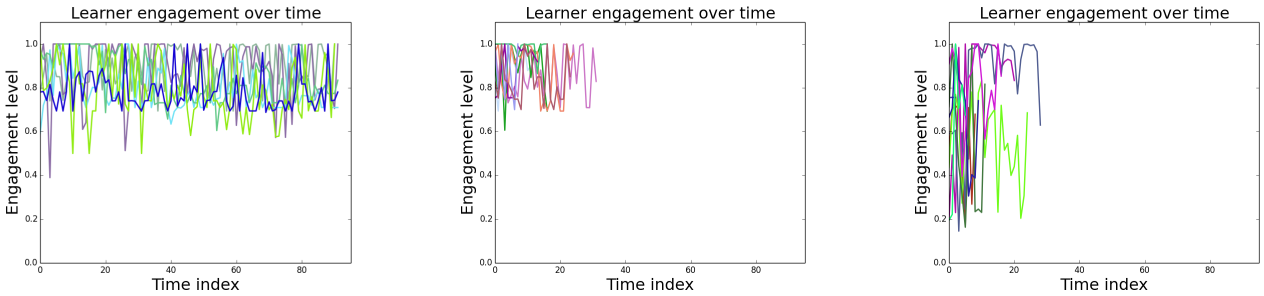
The first two weeks of data ($\Omega^{1,20}$) is sparse in that the majority of learners answer at most a few questions during this time, many of whom will drop out (see Figure 1). In this case, our model obtains a modest improvement over SPARFA, which is static and uses fewer parameters. The gain over BKT is particularly pronounced, at 5.7%. This, combined with the findings for active learners over the full course ($\Omega^{20,92}$), shows that observing video-watching behavior of learners who drop out of the course in its early states (these learners are excluded from $\Omega^{20,92}$) leads to a slight increase in the performance gain of the proposed model over the baseline algorithms. Importantly, this shows that our algorithm provides benefit for *early detection*, with the ability to predict performance of learners who will end up dropping out [8].

4.2 Analyzing engagement

Given predictive quality, one benefit of our model is that it can be used to analyze engagement. The two parameters to consider for this are the regression coefficient vector $\boldsymbol{\beta}$ and the engagement scalar $e_u^{(t)}$ itself.

Behavior and engagement. Table 2 gives each of the estimated feature coefficients in $\boldsymbol{\beta}$ for the full dataset $\Omega^{1,92}$, with regularization parameters chosen via cross validation. All of the features except for the number of fast forwards are positively correlated with the latent engagement level. This is to be expected since many of the features are associated with processing more video content, e.g., spending more time, playing more, or pausing longer to reflect, while fast forwarding involves skipping over the content.

The features that contribute most to high latent engagement levels are the number of pauses, the number of rewinds, and the average playback rate. The first two of these are likely indicators of actual engagement as well, since they indicate whether the learner was thinking while pausing the video or re-visiting earlier content which contains knowledge that they need to recall or revise. The strong, positive correlation of average playback rate is somewhat surprising though: we may expect that a higher playback rate would have a



(a) Learners that consistently exhibit high engagement and finish the course.

(b) Learners that exhibit high engagement but drop out early.

(c) Learners that exhibit inconsistent engagement and drop out.

Figure 4: Plot of the latent engagement level $e_j^{(t)}$ over time for selected learners in three different groups.

negative impact on engagement, like fast forwarding does, as it involves speeding through content. On the other hand, it may be an indication that learners are more focused on the material and trying to keep their interest higher.

Engagement over time. Figure 3 visualizes the evolution of $e_u^{(t)}$ over time for 1/3 of the learners (randomly selected). Patterns in engagement differs substantially across learners; those who finish the course mostly exhibit high engagement levels throughout, while those who drop out early vary greatly in their engagement, some high and others low.

Figure 4 breaks down the learners into three different types according to their engagement patterns, and plots their engagement levels over time separately. The first type of learner (a) finishes the course and consistently exhibits high engagement levels throughout the duration. The second type (b) also consistently exhibits high engagement levels, but drops out of the course after up to three weeks. The third type of learner (c) exhibits inconsistent engagement levels before an early dropout. Equipped with temporal plots like these, an instructor could determine which learners may be in need of intervention, and could design different interventions for different engagement clusters [8, 36].

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new statistical model for learning, based on learner behavior while watching lecture videos and their performance on in-video quiz questions. Our model has two main parts: (i) a response model, which relates a learner's performance to latent concept knowledge, and (ii) a learning model, which relates the learner's concept knowledge in turn to their latent engagement level while watching videos. Through evaluation on a real-world MOOC dataset, we showed that our model can predict unobserved question responses with superior quality to two state-of-the-art baselines, and also that it can lead to engagement analytics: it identifies key behavioral features driving high engagement, and shows how each learner's engagement evolves over time.

Our proposed model enables the measurement of engagement solely from data that is logged within online learning platforms: clickstream data and quiz responses. In this way, it serves as a less invasive alternative to current approaches for measuring engagement that require external devices, e.g., cameras and eye-trackers [6, 16, 35]. One avenue of future work is to conduct an experiment that will correlate our definition of latent engagement with these methods.

Additionally, one could test other, more sophisticated characterizations of the latent engagement variable. One such approach could seek to characterize engagement as a function of learners' previous knowledge level. An alternative or addition to this would be a generative modeling approach of engagement to enable the prediction of future engagement given each learner's learning history.

One of the long-term, end-all goals of this work is the design of a method for useful, real-time analytics to instructors. The true test of this ability comes from incorporating the method into a learning system, providing its outputs – namely, performance prediction forecasts and engagement evolution – to an instructor through the user interface, and measuring the resulting improvement in learning outcomes.

Acknowledgments

Thanks to Debshila Basu Mallick for discussions on the different types of engagement.

6. REFERENCES

[1] Networks: Friends, Money, and Bytes. `https://www.coursera.org/course/friendsmoneybytes`.

[2] L. Allen, M. Jacovina, M. Dascalu, R. Roscoe, K. Kent, A. Likens, and D. McNamara. {ENTER}ing the time series {SPACE}: Uncovering the writing process through keystroke analyses. In *Proc. Intl. Conf. Educ. Data Min.*, pages 22–29, June 2016.

[3] A. Anderson, S. Christenson, M. Sinclair, and C. Lehr. Check & connect: The importance of relationships for promoting engagement with school. *J. School Psychol.*, 42(2):95–113, Mar. 2004.

[4] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Engaging with massive online courses. In *Proc. Intl. Conf. World Wide Web*, pages 687–698, Apr. 2014.

[5] R. Baker, A. Corbett, and K. Koedinger. Detecting student misuse of intelligent tutoring systems. In *Proc. Intl. Conf. Intell. Tutoring Syst.*, pages 531–540, Aug. 2004.

[6] R. Bixler and S. D'Mello. Automatic gaze-based user-independent detection of mind wandering during computerized reading. *User Model. User-adapt. Interact.*, 26(1):33–68, Mar. 2016.

[7] C. Brinton, S. Buccapatnam, F. Wong, M. Chiang, and H. Poor. Social learning networks: Efficiency optimization for MOOC forums. In *Proc. IEEE Conf.*

Comput. Commun., pages 1–9, Apr. 2016.

[8] C. Brinton and M. Chiang. MOOC performance prediction via clickstream data and social learning networks. In *Proc. IEEE Conf. Comput. Commun.*, pages 2299–2307, April 2015.

[9] C. Brinton, R. Rill, S. Ha, M. Chiang, R. Smith, and W. Ju. Individualization for education at scale: MIIC design and preliminary evaluation. *IEEE Trans. Learn. Technol.*, 8(1):136–148, Jan. 2015.

[10] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis – A general method for cognitive model evaluation and improvement. In *Proc. Intl. Conf. Intell. Tutoring Syst.*, pages 164–175, June 2006.

[11] L. Chen, X. Li, Z. Xia, Z. Song, L. Morency, and A. Dubrawski. Riding an emotional roller-coaster: A multimodal study of young child's math problem solving activities. In *Proc. Intl. Conf. Educ. Data Min.*, pages 38–45, June 2016.

[12] C. Coleman, D. Seaton, and I. Chuang. Probabilistic use cases: Discovering behavioral patterns for predicting certification. In *Proc. ACM Conf. Learn at Scale*, pages 141–148, Mar. 2015.

[13] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-adapt. Interact.*, 4(4):253–278, Dec. 1994.

[14] C. Farrington, M. Roderick, E. Allensworth, J. Nagaoka, T. Keyes, D. Johnson, and N. Beechum. *Teaching Adolescents to Become Learners: The Role of Noncognitive Factors in Shaping School Performance–A Critical Literature Review*. Consortium on Chicago School Research, 2012.

[15] B. Gelman, M. Revelle, C. Domeniconi, A. Johri, and K. Veeramachaneni. Acting the same differently: A cross-course comparison of user behavior in MOOCs. In *Proc. Intl. Conf. Educ. Data Min.*, pages 376–381, June 2016.

[16] M. Gilzenrat, J. Cohen, J. Rajkowski, and G. Aston-Jones. Pupil dynamics predict changes in task engagement mediated by locus coeruleus. In *Proc. Soc. Neurosci. Abs.*, page 19, Nov. 2003.

[17] P. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of MOOC videos. In *Proc. ACM Conf. Learn at Scale*, pages 41–50, Mar. 2014.

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[19] H. Jin and C. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.*, 17(3):299–310, Mar. 2005.

[20] J. Kim, P. Guo, D. Seaton, P. Mitros, K. Gajos, and R. Miller. Understanding in-video dropouts and interaction peaks in online lecture videos. In *Proc. ACM Conf. Learn at Scale*, pages 31–40, Mar. 2014.

[21] R. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In *Proc. Intl. Conf. Learn. Analyt. Knowl.*, pages 170–179, Apr. 2013.

[22] K. Koedinger, J. Kim, J. Jia, E. McLaughlin, and N. Bier. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. In *Proc. ACM Conf. Learn at Scale*, pages 111–120, Mar. 2015.

[23] A. Lan, C. Studer, and R. Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In *Proc. ACM SIGKDD Intl. Conf. Knowl. Discov. Data Min.*, pages 452–461, Aug. 2014.

[24] A. Lan, A. Waters, C. Studer, and R. Baraniuk. Sparse factor analysis for learning and content analytics. *J. Mach. Learn. Res.*, 15:1959–2008, June 2014.

[25] L. Malkiewich, R. Baker, V. Shute, S. Kai, and L. Paquette. Classifying behavior to elucidate elegant problem solving in an educational game. In *Proc. Intl. Conf. Educ. Data Min.*, pages 448–453, June 2016.

[26] J. McBroom, B. Jeffries, I. Koprinska, and K. Yacef. Mining behaviours of students in autograding submission system logs. In *Proc. Intl. Conf. Educ. Data Min.*, pages 159–166, June 2016.

[27] Z. Pardos and N. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *Proc. Intl. Conf. User Model. Adapt. Personalization*, pages 255–266, June 2010.

[28] J. Reich, B. Stewart, K. Mavon, and D. Tingley. The civic mission of MOOCs: Measuring engagement across political differences in forums. In *Proc. ACM Conf. Learn at Scale*, pages 1–10, Apr. 2016.

[29] M. San Pedro, E. Snow, R. Baker, D. McNamara, and N. Heffernan. Exploring dynamical assessments of affect, behavior, and cognition and Math state test achievement. In *Proc. Intl. Conf. Educ. Data Min.*, pages 85–92, June 2015.

[30] C. Shi, S. Fu, Q. Chen, and H. Qu. VisMOOC: Visualizing video clickstream data from massive open online courses. In *IEEE Pacific Visual. Symp.*, pages 159–166, Apr. 2015.

[31] S. Slater, R. Baker, J. Ocumpaugh, P. Inventado, P. Scupelli, and N. Heffernan. Semantic features of Math problems: Relationships to student learning and engagement. In *Proc. Intl. Conf. Educ. Data Min.*, pages 223–230, June 2016.

[32] S. Tomkins, A. Ramesh, and L. Getoor. Predicting post-test performance from online student behavior: A high school MOOC case study. In *Proc. Intl. Conf. Educ. Data Min.*, pages 239–246, June 2016.

[33] A. Vail, J. Wiggins, J. Grafsgaard, K. Boyer, E. Wiebe, and J. Lester. The affective impact of tutor questions: Predicting frustration and engagement. In *Proc. Intl. Conf. Educ. Data Min.*, pages 247–254, June 2016.

[34] X. Wang, D. Yang, M. Wen, K. Koedinger, and C. Rosé. Investigating how student's cognitive behavior in MOOC discussion forums affect learning gains. In *Proc. Intl. Conf. Educ. Data Min.*, pages 226–233, June 2015.

[35] J. Whitehill, Z. Serpell, Y. Lin, A. Foster, and J. Movellan. The faces of engagement: Automatic recognition of student engagement from facial expressions. *IEEE Trans. Affect. Comput.*, 5(1):86–98, Jan. 2014.

[36] J. Whitehill, J. Williams, G. Lopez, C. Coleman, and J. Reich. Beyond prediction: Towards automatic intervention in MOOC student stop-out. In *Proc. Intl. Conf. Educ. Data Min.*, pages 171–178, June 2015.

[37] D. Yang, R. Kraut, and C. Rosé. Exploring the effect of student confusion in massive open online courses. *J. Educ. Data Min.*, 8(1):52–83, 2016.

Efficient Feature Embeddings for Student Classification with Variational Auto-encoders

Severin Klingler
Dept. of Computer Science
ETH Zurich, Switzerland
kseverin@inf.ethz.ch

Rafael Wampfler
Dept. of Computer Science
ETH Zurich, Switzerland
wrafael@inf.ethz.ch

Tanja Käser
Graduate School of Education
Stanford University, USA
tkaeser@stanford.edu

Barbara Solenthaler
Dept. of Computer Science
ETH Zurich, Switzerland
sobarbar@inf.ethz.ch

Markus Gross
Dept. of Computer Science
ETH Zurich, Switzerland
grossm@inf.ethz.ch

ABSTRACT

Gathering labeled data in educational data mining (EDM) is a time and cost intensive task. However, the amount of available training data directly influences the quality of predictive models. Unlabeled data, on the other hand, is readily available in high volumes from intelligent tutoring systems and massive open online courses. In this paper, we present a semi-supervised classification pipeline that makes effective use of this unlabeled data to significantly improve model quality. We employ deep variational auto-encoders to learn efficient feature embeddings that improve the performance for standard classifiers by up to 28% compared to completely supervised training. Further, we demonstrate on two independent data sets that our method outperforms previous methods for finding efficient feature embeddings and generalizes better to imbalanced data sets compared to expert features. Our method is data independent and classifier-agnostic, and hence provides the ability to improve performance on a variety of classification tasks in EDM.

Keywords

semi-supervised classification, variational auto-encoder, deep neural networks, dimensionality reduction

1. INTRODUCTION

Building predictive models of student characteristics such as knowledge level, learning disabilities, personality traits or engagement is one of the big challenges in educational data mining (EDM). Such detailed student profiles allow for a better adaptation of the curriculum to the individual needs and is crucial for fostering optimal learning progress. In order to build such predictive models, smaller-scale and controlled user studies are typically conducted where detailed information about student characteristics are at hand (labeled data). The quality of the predictive models, however, inherently depends on the number of study participants, which is typically on the lower side due to time and budget constraints. In contrast to such controlled user studies, digital learning environments such as intelligent tutoring systems (ITS), educational games, learning simulations, and massive open online courses (MOOCs) produce high volumes of data. These data sets provide rich information about student interactions with the system, but come with no or only little additional information about the user (unlabeled data).

Semi-supervised learning bridges this gap by making use of patterns in bigger unlabeled data sets to improve predictions on smaller labeled data sets. This is also the focus of this paper. These techniques are well explored in a variety of domains and it has been shown that classifier performance can be improved for, e.g., image classification [15], natural language processing [28] or acoustic modeling [21]. In the education community, semi-supervised classification has been used employing self-training, multi-view training and problem-specific algorithms. Self-training has e.g. been applied for problem-solving performance [22]. In self-training, a classifier is first trained on labeled data and is then iteratively retrained using its most confident predictions on unlabeled data. Self-training has the disadvantage that incorrect predictions decrease the quality of the classifier. Multi-view training uses different data views and has been explored with co-training [27] and tri-training [18] for predicting prerequisite rules and student performance, respectively. The performance of these methods, however, largely depends on the properties of the different data views, which are not yet fully understood [34]. Problem-specific semi-supervised algorithms have been used to organize learning resources in the web [19], with the disadvantage that they cannot be directly applied for other classification tasks.

Recently, it has been shown (outside of the education context) that variational auto-encoders (VAE) have the potential to outperform the commonly used semi-supervised classification techniques. VAE is a neural network that includes an encoder that transforms a given input into a typically lower-dimensional representation, and a decoder that reconstructs the input based on the latent representation. Hence, VAEs learn an efficient feature embedding (feature representation) using unlabeled data that can be used to improve the performance of any standard supervised learning algorithm [15]. This property greatly reduces the need for problem-specific algorithms. Moreover, VAEs feature the advantage that the trained deep generative models are able to produce realistic samples that allow for accurate data imputation and simulations [23], which makes them an appealing choice for EDM. Inspired by these advantages, and the demonstrated superior classifier performance in other domains as in computer vision [16, 23], this paper explores VAE for student classification in the educational context.

We present a complete semi-supervised classification pipeline that employs deep VAEs to extract efficient feature embeddings from unlabeled student data. We have optimized the architecture of two different networks for educational data - a simple variational auto-encoder and a convolutional variational auto-encoder. While our method is generic and hence widely applicable, we apply the pipeline to the problem of detecting students suffering from developmental dyscalculia (DD), which is a learning disability in arithmetics. The large and unlabeled data set at hand consists of student data of more than 7K students and we evaluate the performance of our pipeline on two independent small and labeled data sets with 83 and 155 students. Our evaluation first compares the performance of the two networks, where our results indicate superiority of the convolutional VAE. We then apply different classifiers to both labeled data sets, and demonstrate not only improvements in classification performance of up to 28% compared to other feature extraction algorithms, but also improved robustness to class imbalance when using our pipeline compared to other feature embeddings. The improved robustness of our VAE is especially important for predicting relatively rare student conditions - a challenge that is often met in EDM applications.

2. BACKGROUND

In the semi-supervised classification setting we have access to a large data set $\mathcal{X}_B$ without labels and a much smaller labeled data set $\mathcal{X}_S$ with labels $\mathcal{Y}_S$. The idea behind semi-supervised classification is to make use of patterns in the unlabeled data set to improve the quality of the classifier beyond what would be possible with the small data set $\mathcal{X}_S$ alone. There are many different approaches to semi-supervised classification including transductive SVMs, graph-based methods, self-training or representation learning [35]. In this work we focus on learning an efficient encoding $\mathbf{z} = E(\mathbf{x})$ for $\mathbf{x} \in \mathcal{X}_B$ of the data domain using the unlabeled data $\mathcal{X}_B$ only. This learnt data transformation $E(\cdot)$ - the encoding - is then applied to the labeled data set $\mathcal{X}_S$. Well-known encoders include principle component analysis (PCA) or Kernel PCA (KPCA). PCA is a dimensionality reduction method that finds the optimal linear transformation from an N-dimensional to a K-dimensional space (given a mean-squared error loss). Kernel PCA [24] extends PCA allowing non-linear transformations into a K-dimensional space and has, among others, been successfully used for novelty detection in non-linear domains [11]. Recently, variational auto-encoders (VAE) have outperformed other semi-supervised classification techniques on several data sets [15]. VAE combine variational inference networks with generative models parametrized by deep neural networks that exploit information in the data density to find efficient lower dimensional representations (feature embeddings) of the data.

Auto-encoder. An auto-encoder or autoassociator [2] is a neural network that encodes a given input into a (typically lower dimensional) representation such that the original input can be reconstructed approximately. The auto-encoder consists of two parts. The encoder part of the network takes the N-dimensional input $\mathbf{x} \in \mathbb{R}^N$ and computes an encoding $\mathbf{z} = E(\mathbf{x})$ while the decoder D reconstructs the input based on the latent representation $\hat{\mathbf{x}} = D(\mathbf{z})$. If we train a network using the mean squared error loss and the network consists of a single linear hidden layer of size K, e.g. $E(\mathbf{x}) = \mathbf{W_1}\mathbf{x} + \mathbf{b_1}$ and $D(\mathbf{z}) = \mathbf{W_2}\mathbf{z} + \mathbf{b_2}$ for weights $\mathbf{W}_1 \in \mathbb{R}^{K \times N}$ and $\mathbf{W}_2 \in \mathbb{R}^{N \times K}$ and offsets $\mathbf{b}_1 \in \mathbb{R}^K$ and $\mathbf{b}_2 \in \mathbb{R}^N$, the autoencoder behaves similar to PCA in that the network learns to project the input into the span of the K first principle components [2]. For more complex networks with non-linear layers multi-modal aspects of the data can be learnt. Auto-encoders can be used in semi-supervised classification tasks because the encoder can compute a feature representation $\mathbf{z}$ of the original data $\mathbf{x}$. These features can then be used to train a classifier. The learnt feature embedding facilitates classification by clustering related observations in the computed latent space.

Variational auto-encoder. Variational auto-encoders [15] are generative models that combine Bayesian inference with deep neural networks. They model the input data $\mathbf{x}$ as

$$p_\theta(\mathbf{x}|\mathbf{z}) = f(\mathbf{x}; \mathbf{z}, \theta) \qquad p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, I) \tag{1}$$

where f is a likelihood function that performs a non-linear transformation with parameters θ of $\mathbf{z}$ by employing a deep neural network. In this model the exact computation of the posterior $p_\theta(\mathbf{z}|\mathbf{x})$ is not computationally tractable. Instead, the true posterior is approximated by a distribution $q_\phi(\mathbf{z}|\mathbf{x})$ [16]. This inference network $q_\phi(\mathbf{z}|\mathbf{x})$ is parametrized as a multivariate normal distribution as

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mu_\phi(\mathbf{x}), \mathrm{diag}(\sigma^2_\phi(\mathbf{x}))), \tag{2}$$

where $\mu_\phi(\mathbf{x})$ and $\sigma^2_\phi(\mathbf{x})$ denote vectors of means and variance respectively. Both functions $\mu_\phi(\cdot)$ and $\sigma^2_\phi(\cdot)$ are represented as deep neural networks. Hence, variational autoencoders essentially replace the deterministic encoder $E(\mathbf{x})$ and decoder $D(\mathbf{z})$ by a probabilistic encoder $q_\phi(\mathbf{z}|\mathbf{x})$ and decoder $p_\theta(\mathbf{x}|\mathbf{z})$. Direct maximization of the likelihood is computationally not tractable, therefore a lower bound on the likelihood has been derived [16]. The learning task then amounts to maximizing this variational lower bound

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right], \tag{3}$$

where KL denotes the Kullback-Leibler divergence. The lower bound consists of two intuitive terms. The first term is the reconstruction quality while the second one regularizes the latent space towards the prior $p(\mathbf{z})$. We perform optimization of this lower bound by applying a stochastic optimization method using gradient back-propagation [14].

3. METHOD

In the following we introduce two networks. First, a simple variational auto-encoder consisting of fully connected layers to learn feature embeddings of student data. These encoders have shown to be powerful for semi-supervised classification [15], and are often applied due to their simplicity. Second, an advanced auto-encoder that combines the advantages of VAE with the superiority of asymmetric encoders. This is motivated by the fact that asymmetric auto-encoders have shown superior performance and more meaningful feature representations compared to simple VAE in other domains such as image synthesis [29].

Student snapshots. There are many applications where we want to predict a label y_n for each student n within an ITS based on behavioral data X_n. These labels typically relate to external variables or properties of a student, such

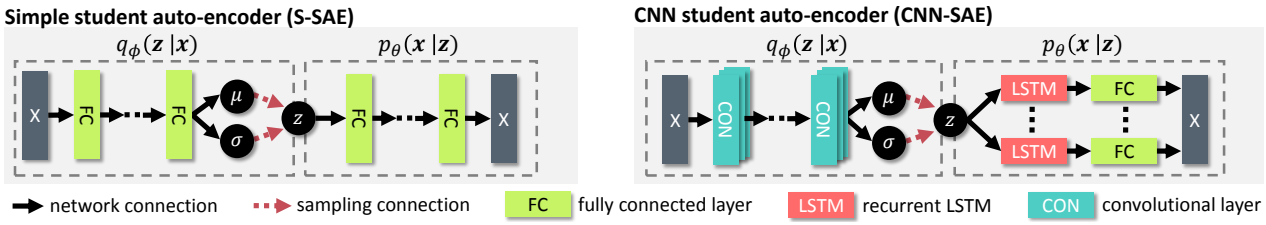


Figure 1: Network layouts for our simple student auto-encoder (left) using only fully connected layers and our improved CNN student auto-encoder (right) using convolutions for the encoder and recurrent LSTM layers for the decoder. In contrast to standard auto-encoders, the connections to the latent space z are sampled (red dashed arrows) from a Gaussian distribution.

as age, learning disabilities, personality traits, learner types, learning outcome etc. Similar to Knowledge Tracing (KT) we propose to model the data $X_n = \{\mathbf{x}_{n1}, \ldots, \mathbf{x}_{nT}\}$ as a sequence of T observations. In contrast to KT we store F different feature values $\mathbf{x}_{nt} \in \mathbb{R}^F$ for each element in the sequence, where t denotes the t^{th} opportunity within a task. This allows us to simultaneously store data from multiple tasks in $\mathbf{x}_{nt}$, e.g. $\mathbf{x}_{n1}$ stores all features for student n that were observed during the first task opportunities. For every task in an ITS we can extract various different features that characterize how a student n was approaching the task. These features include performance, answer times, problem solving strategies, etc. We combine this information into a student snapshot $\mathbf{X}_n \in \mathbb{R}^{T\times F}$, where T is the number of task opportunities and F is the number of extracted features.

Simple student auto-encoder (S-SAE). Our simple variational autoencoder is following the general design outlined in Section 2 and is based on the student snapshot representation. For ease of notation we use $\mathbf{x} := \text{vec}(\mathbf{X_n})$, where $\text{vec}(\cdot)$ is the matrix vectorization function to represent the student snapshot of student n. The complete network layout is depicted in Figure 1, left. The encoder and decoder networks consist of L fully connected layers that are implemented as an affine transformation of the input followed by a non-linear activation function $\beta(\cdot)$ as $\mathbf{x}_l = \beta(\mathbf{W}_l\mathbf{x}_{l-1}+\mathbf{b}_l)$, where l is the layer index and $\mathbf{W}_l$ and $\mathbf{b}_l$ are a weight matrix and offset vector of suitable dimensions. Typical choices for $\beta(\cdot)$ include tanh, rectified linear units or sigmoid functions [6]. To produce latent samples $\mathbf{z}$ we sample from the normal distribution (see Equation (2)) using re-parametrization [16]

$$\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x})\epsilon, \tag{4}$$

where $\epsilon \sim \mathcal{N}(0,1)$, to allow for back-propagation of gradients. For $p_\theta(\mathbf{x}|\mathbf{z})$ (see (1)) any suitable likelihood function can be used. We used a Gaussian distribution for all presented examples. Note that the likelihood function is parametrized by the entire (non-linear) decoder network.

The training of variational auto-encoders can be challenging as stochastic optimization was found to set $q_\phi(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$ in all but vanishingly rare cases [3], which corresponds to a local maximum that does not use any information from $\mathbf{x}$. We therefore add a warm-up phase that gradually gives the regularization term in the target function more weight:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \alpha\,\text{KL}\left[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right], \tag{5}$$

where $\alpha \in [0,1]$ is linearly increased with the number of epochs. The warm-up phase has been successfully used for training deep variational auto-encoders [25]. Furthermore, we initialize the weights of the dense layer computing $\log(\sigma^2_\phi(\mathbf{x}))$ to 0 (yielding a variance of 1 at the beginning of the training). This was motivated by our observations that if we employ standard random weight initialization techniques (glorot-norm, he-norm [9]) we can get relatively high initial estimates for the variance $\sigma^2_\phi(\mathbf{x})$, which due to the sampling leads to very unreliable samples $\mathbf{z}$ in the latent space. The large variance in sampled points in the latent space leads to bad convergence properties of the network.

CNN student auto-encoder (CNN-SAE). Following the recent findings in computer vision we present a second, more advanced network that typically outperforms simpler VAE. In [29], for example, these asymmetric auto-encoders resulted in superior reconstruction of images as well as more meaningful feature embeddings. A specific kind of convolutional neural network was combined with an auto-encoder, being able to directly capture low level pixel statistics and hence to extract more high-level feature embeddings.

Inspired by this previous work, we combine an asymmetric auto-encoder (and a decoder that is able to capture low level statistics) with the advantages of variational auto-encoders. Figure 1, right, shows our combined network. We employ multiple layers of one-dimensional convolutions to parametrize the encoder $q_\phi(\mathbf{z}|\mathbf{x})$ (again we assume a Gaussian distribution, see (2)). The distribution is parametrized as follows:

$$\begin{aligned}\mu_\phi(\mathbf{x}) &= \mathbf{W}_\mu\mathbf{h} + \mathbf{b}_\mu\\ \log(\sigma^2_\phi(\mathbf{x})) &= \mathbf{W}_\sigma\mathbf{h} + \mathbf{b}_\sigma\\ \mathbf{h} = \text{conv}_l(\mathbf{x}) &= \beta(\mathbf{W}_l * \text{conv}_{l-1}(\mathbf{x})),\end{aligned}$$

where $*$ is the convolution operator, $\mathbf{W}_l, \mathbf{W}_\mu, \mathbf{W}_\sigma$ are weights of suitable dimensions, $\beta(\cdot)$ is a non-linear activation function and l denotes the layer depth. Further, $\text{conv}_0(\mathbf{x}) = \mathbf{x}$. We keep the standard variational layer (see (4)) while changing the output layer to a recurrent layer using long term short term units (LSTM). Recurrent layers have successfully been used in auto-encoders before, e.g. in [5]. LSTM were very successful for modeling temporal sequences because they can model long and short term dependencies between time steps. Every LSTM unit receives a copy of the sampled points in latent-space, which allows the LSTM network to combine context information (point in the latent

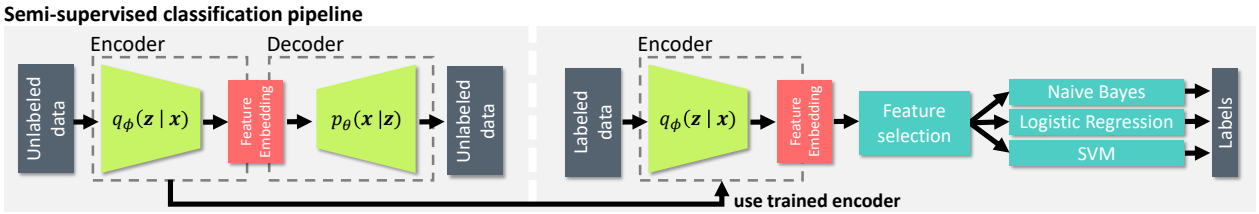


Figure 2: Pipeline overview. We train the variational auto-encoder on a large unlabeled data set. The trained encoder of the auto-encoder can be used to transform other data sets into an expressive feature embedding. Based on this feature embedding we train different classifiers to predict the student labels.

space) with the sequence information (memory unit in the LSTM cell). Using LSTM cells the decoder $p_\theta(\mathbf{x}|\mathbf{z})$ assumes a Gaussian distribution and is parametrized as follows:

$$\mu_{\theta t}(\mathbf{z}) = \mathbf{W}_{\mu z} \cdot \text{lstm}_t(\mathbf{z}) + \mathbf{b}_{\mu z}$$
$$\log(\sigma^2_{\theta t}(\mathbf{z})) = \mathbf{W}_{\sigma z} \cdot \text{lstm}_t(\mathbf{z}) + \mathbf{b}_{\sigma z},$$

where $\mu_{\theta t}(\mathbf{z})$ and $\sigma^2_{\theta t}(\mathbf{z})$ are the t^{th} components of $\mu_\theta(\mathbf{z})$ and $\sigma^2_\theta(\mathbf{z})$, respectively, $\text{lstm}_t(\cdot)$ denotes the t^{th} LSTM cell and $\mathbf{W}_*$ and $\mathbf{b}_*$ denote suitable weight and offset parameters.

Feature selection. VAE provide a natural way for performing feature selection. The inference network $q_\phi(\mathbf{z}|\mathbf{x})$ infers the mean and variance for every dimension z_i. Therefore, the most informative dimension z_i has the highest KL divergence from the prior distribution $p(z_i) = \mathcal{N}(0, 1)$ while uninformative dimensions will have a KL divergence close to 0 [10]. The KL divergence of z_i to $p(z_i)$ is given by

$$KL\left[q_\phi(z_i|\mathbf{x})||p(z_i)\right] = -\log(\sigma_i) + \frac{\sigma_i^2 \mu_i^2}{2} - \frac{1}{2}, \qquad (6)$$

where μ_i and σ_i are the inferred parameter for the Gaussian distribution $q_\phi(z_i|\mathbf{x})$. Feature selection proceeds by keeping the K dimensions z_i with the largest KL divergence.

Semi-supervised classification pipeline. The encoder and the decoder of the variational auto-encoder can be used independently of each other. This independence allows us to take the trained encoder and map new data to the learnt feature embedding. Figure 2 provides an overview of the entire pipeline for semi-supervised classification. In a first unsupervised step we train a VAE on unlabeled data. The learnt encoder $q_\phi(\mathbf{z}|\mathbf{x})$ is then used to transform labeled data sets to the feature embedding. We finally apply our feature selection step that considers the relative importance of the latent dimensions as previously described. We then train standard classifiers (Logistic Regression, Naive Bayes and Support Vector Machine) on the feature embeddings.

4. RESULTS

We evaluated our approach for the specific example of detecting developmental dyscalculia (DD), which is a learning disability affecting the acquisition of arithmetic skills [33]. Based on the learnt feature embedding on a large unlabeled data set the classifier performance was measured on two independent, small and labeled data sets from controlled user studies. We refer to them as *balanced* and *imbalanced* data sets since their distribution of DD and non-DD children differs: the first study has approximately 50% DD, while the second one includes 5% DD (typical prevalence of DD).

4.1 Experimental Setup

All three data sets were collected from *Calcularis*, which is an intelligent tutoring system (ITS) targeted at elementary school children suffering from DD or exhibiting difficulties in learning mathematics [13]. *Calcularis* consists of different games for training number representations and calculation. Previous work identified a set of games that are predictive of DD within *Calcularis* [17]. Since timing features were found to be one of the most relevant indicators for detecting DD [4] and to facilitate comparison to other feature embedding techniques we limited our analysis to log-normalized timing features, for which we can assume normal distribution [30]. Therefore, we evaluated our pipeline on the subset of games from [17] for which meaningful timing features could be extracted and sufficient samples were available in all data sets (we used >7000 samples for training the VAEs). Since our pipeline currently does not handle missing data only students with complete data were included.

Timing features were extracted for the first 5 tasks in 5 different games. The selected games involve addition tasks (adding a 2-digit number to a 1-digit number with ten-crossing; adding two 2-digit numbers with ten-crossing), number conversion (spoken to written numbers in the ranges 0-10 and 0-100) and subtraction tasks (subtracting a 1-digit number from a 2-digit number with ten-crossing). For every task we extracted the total answer time (time between the task prompt until the answer was entered) and the response time (time between the task prompt and the first input by the student). Hence, each student is represented by a 50-dimensional snapshot $\mathbf{x}$ (see Section 3).

Unlabeled data set. The unlabeled data set was extracted using live interaction logs from the ITS *Calcularis*. In total, we collected data from 7229 children. Note that we have no additional information about the children such as DD or grade. We excluded all teacher accounts as well as log files that were < 20KB. Since every new game in *Calcularis* is introduced by a short video during the very first task, we excluded this particular task for all games.

Balanced data set. The first labeled data set is based on log files from 83 participants of a multi-center user study

conducted in Germany and Switzerland, where approximately half of the participants were diagnosed with DD (47 DD, 36 control) [31]. During the study, children trained with *Calcularis* at home for five times per week during six weeks and solved on average 1551 tasks. There were 28 participants in 2^{nd} grade (9 DD, 19 control), 40 children in 3^{rd} grade (23 DD, 17 control), 12 children in 4^{th} grade (12 DD) and 3 children in 5^{th} grade (3 DD). The diagnosis of DD was based on standardized neuropsychological tests [31].

Imbalanced data set. The second labeled data set is from a user study conducted in the classroom of ten Swiss elementary school classes. In total, 155 children participated, and a prevalence of DD of 5% could be detected (8 DD, 147 control). There were 97 children in 2^{nd} grade (3 DD, 94 control) and 58 children in 3^{rd} grade (5 DD, 53 control). The DD diagnosis was computed based on standardized tests assessing the mathematical abilities of the children [32, 7]. During the study, children solved 85 tasks directly in the classroom. On average, children needed 26 minutes to complete the tasks.

Implementation. The unlabeled data set was used to train the unsupervised VAE for extracting compact feature embeddings of the data. Based on the learnt data transformations we evaluated two standard classifiers: Logistic Regression (LR) and Naive Bayes (NB). We restricted our evaluation to simple classification models because we wanted to assess the quality of the feature embedding and not the quality of the classifier. More advanced classifiers typically perform a (sometimes implicit) feature transformation as part of their data fitting procedure. To represent at least one model that performs such an embedding we included Support Vector Machine (SVM) in all our results. All classifier parameters were chosen according to the default values in *scikit-learn*. Note that we have additionally performed randomized cross-validated hyper-parameter search for all classifiers, which, however, resulted in marginal improvements only. Because of that, and to keep the model simple and especially easily reproducible, we use the default parameter set in this work. For Logistic Regression we used L2 regularization with $C = 1$, for Naive Bayes we used Gaussian distributions and for the SVM RBF kernels and data point weights have been set inversely proportional to label frequencies. All results are cross-validated using 30 randomized training-test splits on the unlabeled data (test size 5%). The classification part of the pipeline is additionally cross-validated using 300 label-stratified random training-test splits (test size 20%) to ensure highly reproducible classification results.

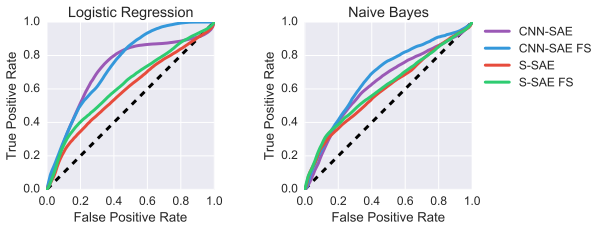
Network hyper-parameters were defined using the approach described in [1]. We increased the number of nodes per layer, the number of layers and the number of epochs until a good fit of the data was achieved. We then regularized the network using dropout [26] with increasing dropout rate until the network was no longer overfitting the data. Activation and weight initialization have been chosen according to common standards: We employ the most common activation function, namely rectified linear activation units (RELU) [20], for all activations. Weight initialization was performed using the method by He et al. [9]. Following this procedure, the following parameters were used for the S-SAE model: encoder and decoders used 3 layers of size 320. The CNN-SAE model was parametrized as follows: 3 convolution layers with 64 convolution kernels and a filter length of 3. We used a single layer of LSTM cells with 80 nodes. We used a batch size of 500 samples and batch normalization and dropout ($r = 0.25$) at every layer. The warm-up phase (see Section 3) was set to 300 epochs. Training was stopped after 1000 (S-SAE) and 500 (CNN-SAE) epochs. The number of latent units was set to 8 in accordance to previous work on detecting students with DD that used 17 features but found that about half of the features were sufficient to detect DD with high accuracy [17]. When feature selection was applied we set the number of features to $K = 4$ and thus we kept exactly half of the latent space features. All networks were implemented using the Keras framework with TensorFlow™ and optimized using Adam stochastic optimization with standard parameters according to [14].

4.2 Performance comparison

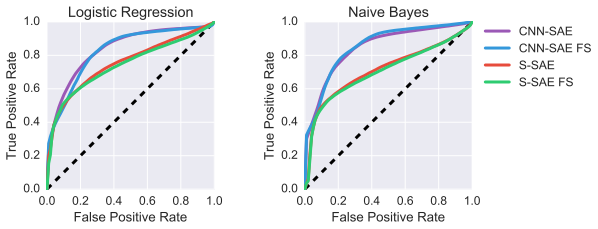
Our VAE models are trained to extract efficient feature embeddings of the data. To assess the quality of these computed feature representations, we compare the classification performance of our method to previous techniques for finding efficient feature embeddings, as well as to feature sets optimized specifically for the task of predicting DD.

Network comparison. In a first experiment we compared the feature embeddings generated by our simple S-SAE and our asymmetric CNN-SAE with and without feature selection. Figure 3 illustrates the average ROC curves of our complete semi-supervised classification pipeline. Our feature embeddings based on asymmetric CNN-SAE clearly outperform the ones from the simple S-SAE on both the imbalanced and the balanced data set for Naive Bayes (NB) and Logistic Regression (LR). For both models, feature selection improves the area under the ROC curve (AUC) for the imbalanced data set (CNN-SAE: LR 4.2%, NB 6.3%; S-SAE: LR 6.8%, NB: 1.6%), but has no effect for the balanced data set. We believe that this is due to the ability of the classifiers to distinguish useful features from noisy ones given enough samples. Since the performance of the classifiers with feature selection (FS) is better or equal to no feature selection in each experiment, we used the CNN-SAE FS model for all further evaluations.

Classification performance. In Figure 4 we compare the classifier performance for different feature embeddings. We compare our method based on VAE to two well-known methods for finding optimal feature embeddings, namely principle component analysis (PCA, green) and Kernel PCA (KPCA, red) [24]. For comparison and as a baseline for the performance of the different methods, we include direct classification results (gray), for which no feature embedding was computed. We used $K = 8$ (dimensionality of feature embedding) for all methods. The features extracted by our pipeline compare favorably to PCA and Kernel PCA showing improvements in terms of AUC of 28% for Logistic Regression and 23% for Naive Bayes on the imbalanced data set and an improvement of 3.75% for Logistic Regression and 7.5% for Naive Bayes on the balanced data set. By using simple classifiers, we demonstrated that our encoder learns an effective feature embedding. More sophisticated classifiers (such as SVM with non-linear kernels) typically proceed by first embedding the input into a specific feature space that is different from the original space.



(a) Imbalanced data set



(b) Balanced data set

Figure 3: ROC curves for the two proposed models with and without feature selection (FS). Our asymmetric CNN-SAE outperforms the simple S-SAE consistently with (blue) and without (purple) feature selection. Feature selection improves performance only on the imbalanced data set.

For the imbalanced data set the overall performance for SVM is significantly lower for all embeddings. This is in line with previous work [12] showing that for imbalanced data sets, the decision boundaries of SVMs are heavily skewed towards the minority class resulting in a preference for the majority class and thus a high miss-classification rate for the minority class. Indeed, we found that SVM predicted only majority labels on the imbalanced data set. For the balanced data set our feature embedding shows improvements of 2.5% over alternative embeddings when using SVM.

Further, Table 1 shows the performance of all feature embeddings using three additional common classification metrics: root mean squared error (RMSE), classification accuracy (Acc.) and area under the precision recall curve (AUPR). We statistically compared the classification metrics of our feature embedding to the best alternative feature embedding using an independent t-test and Bonferroni correction for multiple tests ($\alpha = 0.05$). Our feature embedding significantly outperformed alternative embeddings for all classifiers on both the balanced and imbalanced data sets on most metrics. The main exception was the performance of SVM on the imbalanced data set, which exhibited large variance for all feature embeddings and the worst overall classification performance (compared to the other classifiers).

When comparing classification performance on the imbalanced and the balanced data sets we observed that our pipeline using VAEs showed significant performance improvements compared to other methods for finding feature embeddings. While the unlabeled and the balanced data sets stem from an adaptive version of *Calcularis* the imbalanced data was collected using a fixed task sequence. As our method shows larger improvements on the imbalanced data, we believe CNN-SAE learned an embedding that is robust beyond adaptive ITS. The relative improvements of our feature embeddings is smallest for SVM on the balanced data set. We believe that this is due to ability of the SVM to learn complex decision boundaries given sufficient data. However, the ability for complex decision boundaries renders SVMs more vulnerable to class imbalance, yielding performance at random level on the imbalanced data set.

Comparison to specialized models. Recently, a specialized Naive Bayes classifier (S-NB) for the detection of developmental dyscalculia (DD) was introduced presenting a set of features optimized for the detection of DD [17]. The development of S-NB including the set of features was based on the balanced data set used in this work. In comparison to S-NB, our approach relies on timing data only and the extracted features are independent of the classification task. We compared the performance of S-NB to our CNN-SAE model on both data sets. For the balanced data set we found an AUC of 0.94 for the specialized model (S-NB) compared to an AUC of 0.86 for Naive Bayes using our feature embedding. On the imbalanced data set we found an AUC of 0.67 for S-NB compared to an AUC of 0.77 using Logistic Regression with our feature embedding. These findings demonstrate that while our feature embedding performs slightly worse on the balanced data set (for which the S-NB was developed), we significantly outperform S-NB by 15% on the imbalanced data set, which suggests that our VAE model automatically extracts feature embeddings that are more robust than expert features.

Robustness on sample size. Ideally, a classifier's performance should gracefully decrease as fewer data is provided. A good feature embedding allows a classifier to generalize well based on few labeled examples because similar samples are clustered together in the feature embedding. We therefore investigated the robustness of the different feature representations with respect to the training set size. For this we used the balanced data set where we varied the training set size between 7 (10% of the data) and 62 (90% of the data) by random label-stratified sub-sampling. Figure 5 compares the AUC of the different feature embeddings over different sizes of the training set. In case of Naive Bayes and Logistic Regression our embedding provides superior performance for all training set sizes. For large enough data sets SVM using the raw feature data (Direct, grey) is performing as well as using our embedding (CNN-SAE, blue). However, for smaller data sets starting at 30 samples the performance of SVM based on the raw features declines more rapidly compared to the SVM based on our feature embedding.

5. CONCLUSION

We adapted the recently developed variational auto-encoders to educational data for the task of semi-supervised classification of student characteristics. We presented a complete pipeline for semi-supervised classification that can be used with any standard classifier. We demonstrated that extracted structures from large scale unlabeled data sets can significantly improve classification performance for different labeled data sets. Our findings show that the improvements are especially pronounced for small or imbalanced data sets. Imbalanced data sets typically arise in EDM when detecting relatively rare conditions such as learning disabilities. Im-

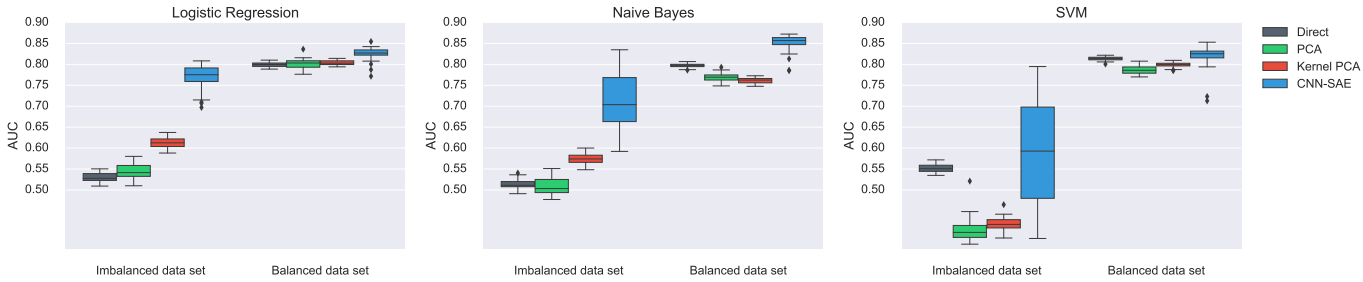


Figure 4: Classification performance for different feature embeddings. Our variational auto-encoder (blue) outperforms other embeddings by up to 28% (imbalanced data set) and by up to 7.5% (balanced data set).

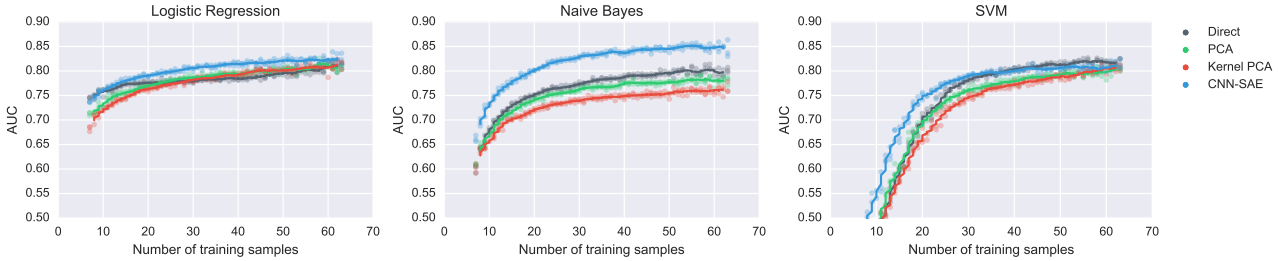


Figure 5: Comparison of classifier performance on the balanced data for different training set sizes (moving average fitted to data points). The features automatically extracted by our variational auto-encoder (blue) maintain a performance advantage even if the training size shrinks to 7 samples (10% of the original size).

Table 1: Comparison of our method to alternative embeddings. Our approach using a variational auto-encoder (CNN-SAE) significantly outperforms other approaches for most cases. The best score for each metric and classifier is shown in bold. *= statistically significant difference (t-test with Bonferroni correction, $\alpha = 0.05$).

	Direct				**PCA**				**Kernel PCA**				**CNN-SAE**			
	AUC	RMSE	AUPR	Acc.	AUC	RMSE	AUPR	Acc.	AUC	RMSE	AUPR	Acc.	AUC	RMSE	AUPR	Acc.
Imbalanced data set																
Logistic Regression	0.53	0.27	0.18	0.91	0.54	0.25	0.17	0.93	0.61	0.25	0.16	0.93	**0.78***	**0.24***	**0.28***	**0.94***
Naive Bayes	0.51	0.29	0.23	0.91	0.50	0.29	0.10	0.90	0.57	0.28	0.20	0.91	**0.70***	**0.25***	**0.24**	**0.93***
SVM	0.55	0.25	**0.22***	0.94	0.40	0.25	0.08	0.94	0.42	0.25	0.09	0.93	**0.59**	0.25	0.16	0.94
Balanced data set																
Logistic Regression	0.80	0.44	0.82	0.73	0.80	0.42	0.84	0.73	0.80	0.42	0.83	0.75	**0.83***	**0.40***	0.84	**0.77**
Naive Bayes	0.80	0.49	0.80	0.73	0.77	0.46	0.77	0.71	0.76	0.46	0.76	0.70	**0.86***	**0.38***	**0.86***	**0.80***
SVM	0.81	0.42	**0.84***	0.75	0.79	0.43	0.81	0.73	0.80	0.43	0.83	0.73	**0.83**	**0.40***	0.81	**0.79***

proved classification results with simple classifiers such as Logistic Regression might indicate that VAEs learn feature embeddings that are interpretable by human experts. In the future we want to explore the learnt representations and compare it to traditional categorizations of students (skills, performance, etc.). Additionally, we want to extend our results to include additional feature types and data reliability indicators to handle missing data. Although we trained our networks on comparatively small sample sizes, the presented method scales (due to mini-batch learning) to much larger data sets (>100K users) allowing the training of more complex VAE. Moreover, the generative model $p_\theta(\mathbf{x}|\mathbf{z})$ that is part of any VAE can be used to produce realistic data samples [29]. Up-sampling of the minority class provides a potential way to improve the decision boundaries for classifiers. In contrast to common up-sampling methods such as ADASYN [8], VAE-based sampling does not require nearest neighbor computations which makes them better applicable to small data sets. Preliminary results for random subsets of the balanced data set showed improvements in AUC by up-sampling based on VAE of 2-3% compared to ADASYN. While we applied our method to the specific case of detecting developmental dyscalculia, the presented pipeline is generic and thus can be applied to any educational data set and used for the detection of any student characteristic.

Acknowledgments. This work was supported by ETH Research Grant ETH-23 13-2.

6. REFERENCES

[1] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. 2012.

[2] Y. Bengio et al. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2009.

[3] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio. Generating Sentences from a Continuous Space. In *Proc. CONLL*, pages 10–21, 2016.

[4] B. Butterworth. *Dyscalculia screener*. Nelson Publishing Company Ltd., 2003.

[5] O. Fabius and J. R. van Amersfoort. Variational recurrent auto-encoders. In *Proc. ICLR*, 2015.

[6] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[7] J. Haffner, K. Baro, P. Parzer, and F. Resch. Heidelberger Rechentest: Erfassung mathematischer Basiskomptenzen im Grundschulalter, 2005.

[8] H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Proc. IJCNN*, pages 1322–1328, 2008.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. ICCV*, pages 1026–1034, 2015.

[10] I. Higgins, L. Matthey, X. Glorot, A. Pal, B. Uria, C. Blundell, S. Mohamed, and A. Lerchner. Early visual concept learning with unsupervised deep learning. *arXiv preprint arXiv:1606.05579*, 2016.

[11] H. Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, pages 863–874, 2007.

[12] T. Imam, K. Ting, and J. Kamruzzaman. z-svm: an svm for improved classification of imbalanced data. *AI 2006: Advances in Artificial Intelligence*, pages 264–273, 2006.

[13] T. Käser, G.-M. Baschera, J. Kohn, K. Kucian, V. Richtmann, U. Grond, M. Gross, and M. von Aster. Design and evaluation of the computer-based training program calcularis for enhancing numerical cognition. *Frontiers in Developmental Psychology*, 2013.

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Proc. ICLR*, 2015.

[15] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Proc. NIPS*, pages 3581–3589, 2014.

[16] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *Proc. ICLR*, 2014.

[17] S. Klingler, T. Käser, A. Busetto, B. Solenthaler, J. Kohn, M. von Aster, and M. Gross. Stealth Assessment in ITS - A Study for Developmental Dyscalculia. In *Proc. ITS*, pages 79–89, 2016.

[18] G. Kostopoulos, S. B. Kotsiantis, and P. B. Pintelas. Predicting Student Performance in Distance Higher Education Using Semi-supervised Techniques. In *Proc. MEDI*, pages 259–270, 2015.

[19] I. Labutov and H. Lipson. Web as a textbook: Curating Targeted Learning Paths through the Heterogeneous Learning Resources on the Web. In *Proc. EDM*, pages 110–118, 2016.

[20] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, pages 436–444, 2015.

[21] H. Liao, E. McDermott, and A. Senior. Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription. In *Proc. ASRU*, pages 368–373, 2013.

[22] W. Min, B. W. Mott, J. P. Rowe, and J. C. Lester. Leveraging semi-supervised learning to predict student problem-solving performance in narrative-centered learning environments. In *Proc. ITS*, pages 664–665, 2014.

[23] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proc. ICML*, pages 1278–1286, 2014.

[24] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *Proc. ICANN*, pages 583–588, 1997.

[25] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In *Proc. NIPS*, pages 3738–3746, 2016.

[26] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, pages 1929–1958, 2014.

[27] V. Tam, E. Y. Lam, S. Fung, W. Fok, and A. H. Yuen. Enhancing educational data mining techniques on online educational resources with a semi-supervised learning approach. In *Proc. TALE*, pages 203–206, 2015.

[28] J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proc. ACL*, pages 384–394, 2010.

[29] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional Image Generation with PixelCNN Decoders. In *Proc. NIPS*, pages 4790–4798, 2016.

[30] W. J. van der Linden. A lognormal model for response times on test items. *Journal of Educational and Behavioral Statistics*, 31(2):181–204, 2006.

[31] M. Von Aster, L. Rauscher, K. Kucian, T. Käser, U. McCaskey, and J. Kohn. Calcularis - Evaluation of a computer-based learning program for enhancing numerical cognition for children with developmental dyscalculia, 2015. 62nd Annual Meeting of the American Academy of Child and Adolescent Psychiatry.

[32] M. von Aster, M. W. Zulauf, and R. Horn. *Neuropsychologische Testbatterie für Zahlenverarbeitung und Rechnen bei Kindern: ZAREKI-R*. Pearson, 2006.

[33] M. G. Von Aster and R. S. Shalev. Number development and developmental dyscalculia. *Developmental Medicine & Child Neurology*, pages 868–873, 2007.

[34] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *Neural Comput. Appl.*, pages 2031–2038, 2013.

[35] X. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison, 2006.

Predicting Short- and Long-Term Vocabulary Learning via Semantic Features of Partial Word Knowledge

SungJin Nam
School of Information
University of Michigan
Ann Arbor, MI 48109
sjnam@umich.edu

Gwen Frishkoff
Department of Psychology
University of Oregon
Eugene, OR 97403
gfrishkoff@gmail.com

Kevyn Collins-Thompson
School of Information
University of Michigan
Ann Arbor, MI 48109
kevynct@umich.edu

ABSTRACT

We show how the novel use of a semantic representation based on Osgood's semantic differential scales can lead to effective features in predicting short- and long-term learning in students using a vocabulary learning system. Previous studies in students' intermediate knowledge states during vocabulary acquisition did not provide much information on which semantic knowledge students gained during word learning practice. Moreover, these studies relied on human ratings to evaluate the students' responses. To solve this problem, we propose a semantic representation for words based on Osgood's semantic decomposition of vocabulary [16]. To demonstrate our method can effectively represent students' knowledge in vocabulary acquisition, we build models for predicting the student's short-term vocabulary acquisition and long-term retention. We compare the effectiveness of our Osgood-based semantic representation to that provided by Word2Vec neural word embedding [13], and find that prediction models using features based on Osgood scale-based scores (OSG) perform better than the baseline and are comparable in accuracy to those using Word2Vec score-based models (W2V). By using more interpretable Osgood-based scales, our study results can help with better understanding of students' ongoing learning states and designing personalized learning systems that can address an individual's weak points in vocabulary acquisition.

Keywords

Vocabulary learning, semantic similarity, prediction model, intelligent tutoring system

1. INTRODUCTION

Studies of word learning have shown that knowledge of individual words is typically not all-or-nothing. Rather, people acquire varying degrees of knowledge of many words incrementally over time, by exposure to them in context [9]. This is especially true for so-called "academic" words that are less common and more abstract — e.g., *pontificate*, *probity*, or *assiduous* [7]. Binary representations and measures model word knowledge simply as correct or incorrect on a particular item (word), but in reality, a student's knowledge level may reside between these two extremes. Thus, previous studies of vocabulary acquisition have suggested that students' partial knowledge be modeled using a representation that adding an additional label corresponding to an intermediate knowledge state [6] or further, in terms of continuous metrics for semantic similarity [3].

In addition, there are multiple dimensions to a word's meaning [16]. Measuring a student's partial knowledge on a single scale may only provide abstract information about the student's general answer quality and not give enough information to specify *which* dimensions of word knowledge a student already has learned or needs to improve. In order to achieve detailed understanding of a student's learning state, online learning systems should be able to capture a student's "learning trajectory" that tracks their partial knowledge on a particular item over time, over multiple dimensions of meaning in a multidimensional semantic representation.

Hence, multidimensional representations of word knowledge can be an important element for building an effective intelligent tutoring system (ITS) for reading and language. Maintaining a fine-grained semantic representation of a student's degree of word knowledge can be helpful for the ITS to design more engaging instructional content, more helpful personalized feedback, and more sensitive assessments [17, 19]. Selecting semantic representations to model, understand, and predict learning outcomes is important to designing a more effective and efficient ITS.

In this paper, we explore the use of multidimensional semantic word representations for modeling and predicting short- and long-term learning outcomes in a vocabulary tutoring system. Our approach derives predictive features using a novel application of existing methods in cognitive psychology combined with methods from natural language processing (NLP). First, we introduce a new multidimensional representation of a word based on the Osgood semantic differential [16], an empirically based, cognitive framework that uses a small number of scales to represent latent components of word meaning. We compare the effectiveness of model features based on this Osgood-based representation to features based on a different representation, the widely-used Word2Vec word embedding [13]. Second, we evaluate our prediction models using data from a meaning-generation task that was conducted during a computer-based intervention. Our study results demonstrate how similarity-based metrics based on rich

semantic representation can be used to automatically evaluate specific components of word knowledge, track changes in the student's knowledge toward the correct meaning, and compute a rich set of features for use in predicting short- and long-term learning outcomes. Our methods could support advances in real-time, adaptive support for word semantic learning, resulting in more effective personalized learning systems.

2. RELATED WORK

The present study is informed by three areas of research: (1) studies of partial word knowledge; (2) the Osgood framework for multiple dimensions of word meaning, and (3) computational methods for estimating semantic similarity.

Partial Word Knowledge. The concept of partial word knowledge has interested vocabulary researchers for several decades, particularly in the learning and instruction of "Tier 2" words [20]. Tier 2 words are low-frequency and typically have complex (multiple, nuanced) meanings. By nature, they are rarely learned through "one-shot" learning or direct definition. Instead, they are learned partially and gaps are filled in over time.

Words in this intermediate state, neither novel nor fully known, are sometimes called "frontier words" [5]. Durso and Shore operationalized the frontier word as a word the student had seen previously but was not actively using it [6]. Based on this definition, the student may have had implicit memory of frontier words, such as general information like whether the word indicates a good or bad situation or refers a person or an action. They discovered that students are more familiar with frontier words than other types of words in terms of their sounds and orthographic characteristics [6]. This previous work suggested that the concept of frontier words can be used to represent a student's partial knowledge states in a vocabulary acquisition task [5, 6].

In some studies, partial word knowledge has been represented using simple, categorical labels, e.g., multiple-choice tests that include "partially correct" response options, as well as a single "best" (correct) response. In other studies, the student is presented with a word and is asked to say what it means [1]. The definition is given partial credit if it reflects knowledge that is partial or incomplete. For example, a student may recognize that the word *probity* has a positive connotation, even if she cannot give a complete definition. However, single categorical or score-based indicators may not explain which specific aspects of vocabulary knowledge the student is missing. Moreover, these studies relied on human ratings to evaluate students' responses for unknown words [6]. Although widely used in psychometric and psycholinguistic studies [4, 16], hiring human raters is expensive and may not be done in real time during students' interaction with the tutoring system.

To address these problems, we propose a data-driven method that can automatically extract semantic characteristics of a word based on a set of relatively simple, interpretable scales. The method benefits from existing findings in cognitive psychology and natural language processing. In the following sections, we illustrate more details of related findings and how they can be used in an intelligent tutoring system setting.

Semantic Representation & the Osgood Framework. To quantify the semantic characteristics of a student's intermediate knowledge of vocabulary, this paper uses a "spatial analogue" for capturing semantic characteristics of words. In [16], Osgood investigated how the meaning of a word can be represented by a series of general semantic scales. By using these scales, Osgood suggested that the meanings of any word can be projected and explored in a continuous semantic space.

Osgood asked human raters to evaluate a set of words using a large number of scales (e.g., tall-short, fat-thin, heavy-light) and captured the semantic representation of a word [16]. Respondents gave Likert ratings, which indicated whether they thought that a word meaning was closer to one extreme (-3) or the other (+3), or basically irrelevant (0). A principal components analysis (PCA) was used to represent the latent semantic features that can explain the patterns of response to individual words within this task.

In our study, we suggest a method that can automatically extract similar semantic information that can project a word into a multidimensional semantic space. By using semantic scales selected from [16], we verify if such representation of semantic attributes of words is useful for predicting students' short- and long-term learning.

Semantic Similarity Measures. Studies in NLP have suggested methods to automatically evaluate the semantic association between two words. For example, Markov Estimation of Semantic Association (MESA) [3, 9] can estimate the similarity between words from a random walk model over a synonym network such as WordNet [14]. Other methods like latent semantic analysis (LSA) are based on co-occurrence of the word in a document corpus. In LSA, semantic similarity between words is determined by using a cosine similarity measure, derived from a sparse matrix constructed from unique words and paragraphs containing the words [10].

For this paper, we use Word2Vec [13], a widely used word embedding method, to calculate the semantic similarity between words. Word2Vec's technique [11] transforms the semantic context, such as proximity between words, into a numeric vector space. In this way, linguistic regularities and patterns are encoded into linear translations. For example, using outputs from Word2Vec, relationships between words can be estimated by simple operations on their corresponding vectors, e.g., *Madrid - Spain + France = Paris*, or *Germany + capital = Berlin* [13].

Measures from these computational semantic similarity tools are powerful because they can provide an automated method for evaluation of partial word knowledge. However, they typically produce a single measure (e.g., cosine similarity or Euclidean distance), representing semantic similarity as a one-dimensional construct. With such a measure, it is not possible to determine represent partial semantic knowledge and changes in knowledge of latent semantic features as word knowledge progresses from unknown to frontier to fully known. In following sections, we describe how we address this problem, using novel methods to to estimate the contribution of Osgood semantic features to individual word meanings.

2.1 Overview of the Study

Based on findings from existing studies, this study will suggest an automatized method for evaluating students' partial knowledge of vocabulary that can be used to predict students' short-term vocabulary acquisition and long-term retention. To investigate this problem, we will answer the following research questions with this paper.

The first research question (RQ1): Can semantic similarity scores from Word2Vec be used to predict students' short-term learning and long-term retention? Previous studies in vocabulary tutoring systems tend to focus on how different experimental conditions, such as different spacing between question items [18], difficulty levels [17], and systematic feedback [7], affect students' short-term learning. This study will answer how computationally estimated trial-by-trial scores in a vocabulary tutoring system can be used to predict students' short-term learning and long-term retention.

RQ2: Compared to using regular Word2Vec scores, how does the model using Osgood's semantic scales [16] as features perform for immediate and delayed learning prediction tasks? As described in the previous section, the initial outcome from Word2Vec returns hundreds of semantic dimensions to represent the semantic characteristics of a word. Summary statistics for comparing such high-dimensional vectors, such as cosine similarity or Euclidean distance, only provide the overall similarity between words. If measures from Osgood scales work in a similar level to models using regular Word2Vec scores for predicting students' learning outcomes, we can argue that it can be an effective method for representing students' partial knowledge of vocabulary.

3. METHOD

3.1 Word Learning Study

This study used a vocabulary tutoring system called Dynamic Support of Contextual Vocabulary Acquisition for Reading (DSCoVAR) [8]). DSCoVAR aims to support efficient and effective learning vocabulary in context. All participants accessed DSCoVAR in a classroom-setting environment by using Chromebook devices or the school's computer lab in the presence of other students.

3.1.1 Study Participants

Participants included 280 middle school students (6th to 8th grade) from multiple schools, including children from diverse socio-economic and educational backgrounds. Table 1 provides a summary of student demographics, including location (P1 or P2), age and grade level, sex. Location P1 is a laboratory school affiliated with a large urban university in the northeastern United States. Students from location P1 were generally of high socio-economic status (e.g., children of University faculty and staff). Location P2 includes three public middle schools in a southern metropolitan area of the United States. All students from location P2 qualified for free or reduced lunch. The study included a broad range of students so that the results of this analysis were more likely to generalize to future samples.

3.1.2 Study Materials

DSCoVAR presented students with 60 SAT-level English words (also known as Tier 2 words). These "target words," lesser-known words that the students are going to learn,

Table 1: The number of participants by grade and gender

	6th grade		7th grade		8th grade	
Group	Girl	Boy	Girl	Boy	Girl	Boy
P1	16	28	19	23	18	13
P2	53	51	12	6	21	20

were balanced between different parts of speech, including 20 adjectives, 20 nouns, and 20 verbs. Based on previous works, we expected that students would have varying degrees of familiarity with the words at pre-test, but that most words would be either completely novel ("unknown") or somewhat familiar ("partially known") [8, 15]. This selection of materials ensured that there would be variability in word knowledge across students for each word and across words for each student.

In DSCoVAR, students learned how to infer the meaning of an unknown word in a sentence by using surrounding contextual information. Having more information in a sentence (i.e., a sentence with a high degree of contextual constraint) can decrease the uncertainty of inference. In this study, the degree of sentence constraint was determined using standard cloze testing methods: quantifying the diversity of responses from 30 human judges when the target word is left as a fill-in-the-blank question.

3.1.3 Study Protocol

The word learning study comprised four parts: (1) a pre-test, which was used to estimate baseline knowledge of words, (2) a training session, where learners were exposed to words in meaningful contexts, (3) an immediate post-test, and (4) a delayed post-test, which occurred approximately one week after training.

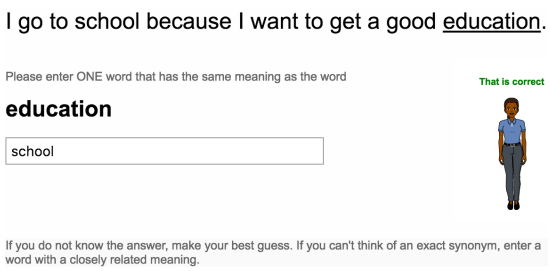
Pre-test. The pre-test session was designed to measure the students' prior knowledge of the target words. For each target word, students were asked to answer two types of questions: familiarity-rating questions and synonym selection questions. In familiarity rating questions, students provided their self-rated familiarity levels (unknown, known, and familiar) for presented target words. In synonym-selection questions, students were asked to select a synonym word for the given target word from five multiple choice options. The outcome from synonym-selection questions provided more objective measures for students' prior domain knowledge of target words.

Training. Approximately one week after the pre-test session, students participated in the training. During training, students learned strategies to infer the meaning of an unknown word in a sentence by using surrounding contextual information.

A training session consisted of two parts: an instruction video and practice questions. In the instruction video, students saw an animated movie clip about how to identify and use contextual information from the sentence to infer the meaning of an unknown word. In the practice question part, students could exercise the skill that they learned from the video. DSCoVAR provided sentences that included a target word with different levels of surrounding contextual information. The amount of contextual information for each sentence was determined by external crowd workers (details described in Section 3.1.2). In the practice question part, each target word was presented four times within

different sentences. Students were asked to type a synonym of the target word, which was presented in the sentence as underlined and bold. Over two weeks, students participated in two training sessions with a week's gap between them. Each training session contained the instruction video and practice questions for 30 target words. An immediate post-test session followed right after each training session.

Figure 1: An example of a training session question. In this example, the target word is "education" with a feedback message for a high-accuracy response.



Students were randomly selected to experience different instruction video conditions (full instruction video vs. restricted instruction video). Additionally, various difficulty level conditions and feedback conditions (e.g., DSCoVAR provides a feedback message to the student based on answer accuracy vs. no feedback) were tested within the same student. However, in this study, we focused on data from students who experienced a full instruction video and repeating difficulty conditions. Repeating difficulty conditions included questions with all high or medium contextual constraint levels. By doing so, we wanted to minimize the impact from various experimental conditions for analyzing post-test outcomes. Moreover, we filtered out response sequences that did not include all four responses for the target word. As a result, we analyzed 818 response sequences from 7,425 items in total.

Immediate and Delayed Post-test. The immediate post-test occurred right after the students finished the training; the delayed post-test was conducted one week later. Data collected during the immediate and delayed post-tests were used to estimate short-and long-term learning, respectively. Test items were identical to those in the pretest session, except for item order, which varied across tests. For analysis of the delayed post-test data, we only used the data from target words for which the student provided a correct answer in the earlier, immediate post-test session. As a result, 449 response sequences were analyzed for predicting the long-term retention.

3.2 Semantic Score-Based Features

We now describe the semantic features tested in our prediction models.

3.2.1 Semantic Scales

For this study, we used semantic scales from Osgood's study [16]. Ten scales were selected by a cognitive psychologist as being considered semantic attributes that can be detected during word learning (Figure 2). Each semantic scale consists of pairs of semantic attributes. For example, the *bad–good* scale can show how the meaning of a word can be projected on a scale with *bad* and *good* located at either

Figure 2: Ten semantic scales used for projecting target words and responses [16].

- bad – good
- complex – simple
- passive – active
- fast – slow
- powerful – helpless
- noisy – quiet
- big – small
- new – old
- helpful – harmful
- healthy – sick

end. The word's relationship with each semantic anchor can be automatically measured from its semantic similarity with these exemplar semantic elements.

3.2.2 Basic Semantic Distance Scores

To extract meaningful semantic information, we have applied the following measures that can be used to explain various characteristics of student responses for different target words. In this study, we used a pre-trained model for Word2Vec,[1] built based on the Google News corpus (100 billion tokens with 3 million unique vocabularies, using a negative sampling algorithm), to measure semantic similarity between words. The output of the pre-trained Word2Vec model contained a numeric vector with 300 hundred dimensions.

First, we calculated the relationship between word pairs (i.e., a single student response and the target word, or a pair of responses) in both the regular Word2Vec (W2V) score and the Osgood semantic scale (OSG) score. In the W2V score, the semantic relationship between words was represented with a cosine similarity between word vectors:

$$D_{w2v}(w_1, w_2) = 1 - |sim(V(w_1), V(w_2))|. \quad (1)$$

In this equation, the function V returned the vectorized representation of the word (w_1 or w_2) from the pre-trained Word2Vec model. By calculating the cosine similarity between two vectors (a cosine similarity function is noted as sim), we could extract a single numeric similarity score between two words. This score was converted into a distance-like score by taking the absolute value of the cosine similarity score and subtracting from one.

For the OSG score, we extracted two different types of scores: a non-normalized score and a normalized score. A non-normalized score showed how a word is similar to a single anchor word (e.g., *bad* or *good*) from the Osgood scale.

$$S^{non}_{osg}(w, a_{i,j}) = sim(V(w), V(a_{i,j})) \quad (2)$$

$$D^{non}_{osg}(w_1, w_2; a_{i,j}) = |S^{non}_{osg}(w_1, a_{i,j})| - |S^{non}_{osg}(w_2, a_{i,j})| \quad (3)$$

In equation 2, $a_{i,j}$ represents a single anchor word (j) in the i-th Osgood scale. The similarity between the anchor word and the evaluating word w was calculated with cosine similarity of Word2Vec outcomes for both words. In a non-normalized setting, the distance between two words given by a particular anchor word was calculated by the difference of absolute cosine similarity scores (equation 3).

The second type of OSG score is a normalized score. By using Word2Vec's ability to do arithmetical calculation of

[1] API and pre-trained model for Word2Vec was downloaded from this URL: https://github.com/3Top/word2vec-api

multiple word vectors, the normalized OSG score provided a relative location of the word from two anchor ends of the Osgood scale.

$$S_{osg}^{nrm}(w, a_i) = sim(V(w), V(a_{i,1}) - V(a_{i,2})) \quad (4)$$

$$D_{osg}^{nrm}(w_1, w_2; a_i) = |S_{osg}^{nrm}(w_1, a_i) - S_{osg}^{nrm}(w_2, a_i)| \quad (5)$$

In equation 4, the output represents the cosine similarity score between the word w and two anchor words ($a_{i,1}$ and $a_{i,2}$). For example, if the cosine similarity score of $S_{osg}^{nrm}(w, a_i)$ is close to -1, it means the word w is close to the first anchor word $a_{i,1}$. If the score is close to 1, it is vice versa. In equation 5, the distance between two words was calculated as the absolute value of the difference between two cosine similarity measures.

3.2.3 Deriving Predictive Features

Based on semantic distance equations explained in the previous section, this section explains examples of predictive features that we used to predict students' short-term learning and long-term retention.

Distance Between the Target Word and the Response. For regular Word2Vec score models and Osgood scale score models, distance measures between the target word and the response (by using equations 1 and 5) were used to estimate the accuracy of the response to a question. This feature represents the trial-by-trial answer accuracy of a student response. Each response sequence for the target word contained four distance scores.

Difference Between Responses. Another feature that we used in both types of models was the difference between responses. This feature could capture how a student's current answer is semantically different from the previous response. From each response sequence, we could extract three derivative scores from four responses.

Convex Hull Area of Responses. Alternative to the difference between responses feature, Osgood scale models were also tested with the area size of convex hull that can be generated by responses calculated with non-normalized Osgood scale scores (equation 3). For example, for each Osgood scale, a non-normalized score provided two-dimensional scores that can be used for geometric representation. By putting the target word in an origin position, a sequence of responses can create a polygon that can represent the semantic area that the student explored with responses. Since some response sequences were unable to generate the polygon by including less than three unique responses, we added a small, random noise that uniformly distributed (between -10^{-4} and 10^{-4}) to all response points. Additionally, a value of 10^{-20} was added to all convex hull area output to create a visible lower-bound value.

Unlike the measure of difference between responses, this feature also considers angles that can be created between responses and the target word. This representation can provide more information than just using difference between responses.

3.3 Modeling

To predict students' short-term learning and long-term retention, we used a mixed-effect logistic regression model (MLR). MLR is a general form of logistic regression model that includes random effect factors to capture variations from repeated measures.

3.3.1 Off-line Variables

Off-line variables capture item- or subject-level variances that can be observed repeatedly from the data. In this study, we used multiple off-line variables as random effect factors.

First, results from familiarity-rating and synonym-selection questions from the pre-test session were used to include item- and subject-level variances. Both variables include information on the student's prior domain knowledge level for target words. Second, the question difficulty condition was considered as an item group level factor. In the analysis, sentences for the target word that were presented to the student contained the same difficulty level, either high or medium contextual constraint levels, over four trials. Third, a different experiment group was used as a subject group factor. As described in Section 3.1.1, this study contains data from students in different institutions in separate geographic locations. The inclusion of these participant groups in the model can be used to explain different short-term learning outcomes and long-term retention by demographic groups.

3.3.2 Model Building

In this study, we compared the performance of MLR models with four different feature types. First, the baseline model was set to indicate the MLR model's performance without any fixed effect variables but only with random intercepts. Second, the response time model was built to be compared with semantic score-based models. Many previous studies have used response time as an important predictor of student engagement and learning [2, 12]. In this study, we used two types of response time variables, the latency for initiating the response and finishing typing the response, as predictive features. Both variables were measured in milliseconds over four trials and natural log transformed for the analysis. Third, semantic features from regular Word2Vec scores were used as predictors. This model was built to show how semantic scores from Word2Vec can be useful for predicting students' short- and long-term performance in DSCoVAR. Lastly, Osgood scale-based features were used as predictors. This model was compared with the regular Word2Vec score model to examine the effectiveness of using Osgood scales for evaluating students' performance in DSCoVAR. For these semantic-score based models, we tested out different types of predictive features that were described in Section 3.2.3. All models shared the same random intercept structure that treated each off-line variable as an individual random intercept.

For Osgood scale models, we also derived reduced-scale models. Reduced-scale models were compared with the full-scale model, which uses all ten Osgood scales. In this case, using fewer Osgood scales can provide easier interpretation of semantic analysis for intelligent tutoring system users.

3.3.3 Model Evaluation

To compare performance between different models, this study used various evaluation metrics, including AUC (an area under the curve score from a response operating characteristic (ROC) curve), F_1 (a harmonic mean of precision and recall), and error rate (a ratio of the number of

misclassified items over total items). 95% confidence interval of each evaluation metric was calculated from the outcome of a ten-fold cross-validation process repeated over ten times.

To select the semantic score-based features for models based on regular Word2Vec scores and Osgood scale scores, we used rankings from each evaluation metric. The model with the highest overall rank (i.e., sum the ranks from AUC, F_1, and error rate, and select the model with the lowest rank-sum value) was considered the best-performing model for the score type (i.e., models based on the regular Word2Vec score or Osgood scale score). More details on this process will be illustrated in the next section.

4. RESULTS

4.1 Selecting Models

In this section, we selected the best-performing model based on the models' overall ranks in each evaluation metric. All model parameters were trained in each fold of repeated cross-validation. We calculated 95% confidence intervals for comparison. To calculate the confidence interval of F_1 and error rate measures, the maximum (F_1) and minimum (error rate) scores of each fold were extracted. These maximum and minimum values were derived from applying multiple cutoff points to the mixed-effect regression model.

4.1.1 Predicting Immediate Learning

First, we built models that predict the students' immediate learning from the immediate post-test session. From models based on regular Word2Vec scores (W2V), the model with the distance between the target and responses and the difference between responses (*Dist+Resp*) provided the highest rank from various evaluation metrics (Table 2). From models based on Osgood scales (OSG), the model with the difference between responses (*Resp*) provided the highest rank.

The selected W2V model provided significantly better performance than the baseline model. The selected OSG model also showed significantly better performance than the baseline model, except for the AUC score. The selected W2V model was significantly better than the model using response time features in the AUC score and error rates.

The selected W2V model showed significantly better performance than the OSG model only with the AUC score. Figure 3 shows that the W2V model has a slightly larger area under the ROC curve than the OSG model. In the precision and recall curve, the selected W2V model provides more balanced trade-offs between precision and recall measures. The selected OSG model outperforms the W2V model in precision only in a very low recall measure range.

4.1.2 Predicting Long-Term Retention

We also built prediction models to predict the students' long-term retention in the delayed post-test session. In this analysis, a student response was included only when the student provided correct answers to the immediate post-test session questions. Among W2V score-based models, the best-performing model contained the same feature types as the immediate post-test results (Table 3). By using the distance between the target and responses and difference between responses (*Dist+Resp*), the model achieved significantly better performance than the baseline model, except for the AUC score.

For OSG models, the model with a convex hull area of responses (*Chull*) provided the highest overall rank from evaluation metrics (Table 3). The results were significantly better than the baseline model, and marginally better than the W2V model. Both selected W2V and OSG models were marginally better than the response time model, except the error rate of the OSG model was significantly better.

In Figure 3, the selected W2V model slightly outperforms the OSG model in mid-range true positive rates, while the OSG model performed slightly better in a higher true positive area. Precision and recall curves show similar patterns to those we observed from the immediate post-test prediction models. The OSG model only outperforms the W2V model in a very low recall value area.

4.1.3 Comparing Models

Compared to the selected W2V model in the immediate post-test condition, the selected W2V model in the delayed post-test retention condition showed a significantly lower AUC score, marginally higher F_1 score, and marginally higher error rate. In terms of OSG models, the selected OSG model for delayed post-test retention showed a significantly better F_1 score and error rates than the selected OSG model in the immediate post-test condition. Based on these results, we can argue that Osgood scale scores can be more useful for predicting student retention in the delayed post-test session than predicting the outcome from the immediate post-test.

In terms of selected feature types, the best-performing OSG models used features based on the difference between responses (*Resp*) or the convex hull area (*Chull*) that was created from the relative location of the responses. On the other hand, selected W2V models used both the distance between the target word and responses and difference between responses (*Dist+Resp*). When we compared both W2V and OSG models using the difference between responses feature, we found that performance is similar in the immediate post-test data. However, the OSG model was significantly better in the delayed post-test data. These results show that Osgood scale scores can be more useful for representing the relationship among response sequences.

4.2 Comparing the Osgood Scales

To identify which Osgood scales are more helpful than others for predicting students' performance, we conducted a scale-wise importance analysis. The results from this section reveal which Osgood scales are more important than others, and how the performance of prediction models with a reduced number of scales is comparable with the full-scale model.

4.2.1 Identifying More Important Osgood Scales

In this section, based on the selected Osgood score model from Section 4.1, we identified the level of contribution for features based on each Osgood scale. For example, the selected OSG model for predicting the immediate post-test data uses the difference between responses in ten Osgood scales as features. In order to diagnose the importance level of the first scale (*bad–good*), we can retrain the model with features based on the nine other scales and compare the

Table 2: Ranks of predictive feature sets for regular Word2Vec models (W2V) and Osgood score models (OSG) in the immediate post-test data. All models are significantly better than the baseline model. (Bold: the selected model with highest overall rank.)

	W2V models			OSG models		
Features	AUC	F_1	Err	AUC	F_1	Err
baseline	0.68 [0.67, 0.69] (5)	0.74 [0.73, 0.74] (5)	0.33 [0.33, 0.34] (5)	0.68 [0.67, 0.69] (5)	0.74 [0.73, 0.74] (5)	0.33 [0.33, 0.34] (7)
RT	0.69 [0.68, 0.70] (4)	0.75 [0.75, 0.76] (3)	0.31 [0.31, 0.32] (4)	0.69 [0.68, 0.70] (2)	0.75 [0.74, 0.76] (2)	0.31 [0.31, 0.32] (2)
Dist	0.72 [0.71, 0.74] (1)	0.76 [0.75, 0.76] (2)	0.29 [0.28, 0.30] (2)	0.67 [0.66, 0.68] (7)	0.73 [0.73, 0.74] (7)	0.33 [0.32, 0.34] (6)
Resp	0.70 [0.69, 0.71] (3)	0.75 [0.74, 0.76] (4)	0.31 [0.30, 0.32] (3)	**0.69 [0.68, 0.70] (1)**	**0.75 [0.75, 0.76] (1)**	**0.31 [0.30, 0.32] (1)**
Chull	NA	NA	NA	0.69 [0.68, 0.70] (3)	0.74 [0.73, 0.75] (4)	0.32 [0.31, 0.33] (4)
Dist+Resp	**0.72 [0.71, 0.73] (2)**	**0.76 [0.75, 0.77] (1)**	**0.29 [0.28, 0.30] (1)**	0.68 [0.67, 0.69] (4)	0.74 [0.73, 0.75] (3)	0.31 [0.31, 0.32] (3)
Dist+Chull	NA	NA	NA	0.67 [0.66, 0.68] (6)	0.74 [0.73, 0.74] (6)	0.33 [0.32, 0.34] (5)

Table 3: Ranks of predictive feature sets for W2V and OSG models in the delayed post-test data. All models are significantly better than the baseline model. (Bold: the selected model with highest overall rank.)

	W2V models			OSG models		
Features	AUC	F_1	Err	AUC	F_1	Err
baseline	0.65 [0.64, 0.67] (5)	0.75 [0.74, 0.76] (5)	0.33 [0.32, 0.34] (5)	0.65 [0.64, 0.67] (5)	0.75 [0.74, 0.76] (7)	0.33 [0.32, 0.34] (7)
RT	0.67 [0.65, 0.68] (3)	0.76 [0.76, 0.77] (4)	0.31 [0.30, 0.32] (3)	0.67 [0.65, 0.68] (3)	0.76 [0.76, 0.77] (5)	0.31 [0.30, 0.32] (5)
Dist	0.66 [0.64, 0.68] (4)	0.77 [0.76, 0.78] (3)	0.31 [0.30, 0.32] (4)	0.66 [0.64, 0.68] (4)	0.78 [0.77, 0.79] (3)	0.30 [0.29, 0.31] (3)
Resp	0.69 [0.67, 0.71] (1)	0.77 [0.76, 0.78] (2)	0.30 [0.29, 0.31] (2)	0.63 [0.61, 0.65] (7)	0.76 [0.75, 0.77] (6)	0.32 [0.31, 0.33] (6)
Chull	NA	NA	NA	**0.69 [0.68, 0.71] (1)**	**0.78 [0.77, 0.79] (2)**	**0.28 [0.27, 0.29] (1)**
Dist+Resp	**0.68 [0.66, 0.70] (2)**	**0.78 [0.77, 0.79] (1)**	**0.30 [0.29, 0.31] (1)**	0.64 [0.62, 0.66] (6)	0.77 [0.76, 0.78] (4)	0.31 [0.29, 0.32] (4)
Dist+Chull	NA	NA	NA	0.69 [0.67, 0.71] (2)	0.78 [0.78, 0.79] (1)	0.29 [0.27, 0.30] (2)

performance of the newly trained model with the existing full-scale model.

In Table 4, we picked the top five scales that were important in individual prediction tasks. We found that *big-small*, *helpful-harmful*, *complex-simple*, and *fast-slow* were commonly important Osgood scales for predicting students' performance in immediate post-test and delayed post-test sessions. Scales like *bad-good* and *passive-active* were only important scales in the immediate post-test prediction. Likewise, *new-old* was an important scale only in the delayed post-test prediction.

Table 4: Scale-wise importance of each Osgood scale. Scales were selected based on the sum of each evaluation metric's rank. (Bold: Osgood scales that were commonly important in both prediction tasks; *: top five scales in each prediction task including tied ranks)

	Imm. post-test				Del. post-test			
Scales	AUC	F_1	Err	All	AUC	F_1	Err	All
bad-good	1	1	1	1*	4	10	4	6
passive-active	2	4	3	2*	8	6	6	7
powerful-helpless	7	9	6	7.5	10	8	10	10
big-small	3	3	4	3*	1	3	2	2*
helpful-harmful	4	6	5	5.5*	2	1	1	1*
complex-simple	8	5	2	5.5*	3	5	7	4.5*
fast-slow	5	2	7	4*	6	4	3	3*
noisy-quiet	6	8	8	7.5	7	9	9	9
new-old	9	7	9	9	5	2	8	4.5*
healthy-sick	10	10	10	10	9	7	5	8

4.2.2 Performance of Reduced Models

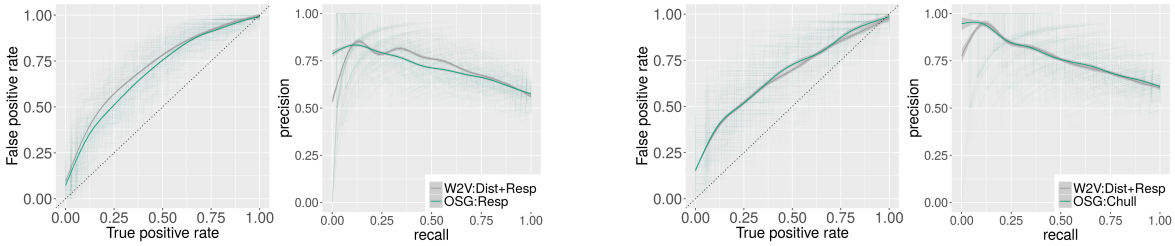
Based on the scale-wise importance analysis results, we built reduced-scale models that only contain features with more important Osgood scales. The prediction performance of reduced-scale models was similar or marginally better than full-scale OSG models. For example, the OSG model for predicting the immediate post-test outcome with the top two scales (*bad–good* and *passive–active*) were marginally better than the full-scale model (AUC: 0.71 [0.70, 0.72], F_1: 0.76 [0.75, 0.77], error rate: 0.30 [0.29, 0.30]). Similar results were observed for predicting retention in the delayed post-test (selected scales: *helpful–harmful*, *big–small*) (AUC: 0.71 [0.69, 0.72], F_1: 0.79 [0.78, 0.80], error rate: 0.28 [0.27, 0.29]). Although differences were small, the results indicate that using a small number of Osgood scales can be similarly effective to the full-scale model.

5. DISCUSSION AND CONCLUSIONS

In this paper, we introduced a novel semantic similarity scoring method that uses predefined semantic scales to represent the relationship between words. By combining Osgood's semantic scales [16] and Word2Vec [13], we could automatically extract the semantic relationship between two words in a more interpretable manner. To show this method can effectively represent students' knowledge in vocabulary acquisition, we built prediction models that can be used to predict the student's immediate learning and long-term retention. We found that our models performed significantly better than the baseline and the response-time-based models. In the future, we believe results from using an Osgood scale-based student model could be used to provide a more personalized learning experience, such as generating questions that can improve an individual student's understanding for specific semantic attributes.

Based on our findings, we have identified the following points for further discussion. First, in Section 4.1, we found that models using Osgood scale scores perform similarly with models using regular Word2Vec scores for predicting students' long-term retention of acquired vocabulary. However, we think our models can be further improved by incorporating additional features. For example, non-semantic score-based features like response time and orthographic similarity among responses can be useful features for capturing different patterns of false predictions of current models. Moreover, some general measures to capture a student's meta-cognitive or linguistic skills could be helpful to explain different retention results found even if students provided the same response sequences. Similarly, in Section 4.1.3, we found that Osgood scores can be a better metric to characterize the relationship between responses in terms of predicting students' retention. A composite model that uses both regular Word2Vec score-based feature (target-response distance) and Osgood scale score-based feature (response-response distance) may also provide better

Figure 3: ROC curves and precision and recall curves for selected immediate post-test prediction models (left) and delayed post-test prediction models (right). Curves are smoothed out with a local polynomial regression method based on repeated cross-validation results.



prediction performance.

Second, we found that models with a reduced number of Osgood scales performed marginally better than the full-scale model. However, differences were very small. Since this study only used some of the semantic scales from Osgood's study [16], further investigation would be required to examine the validity of these scales, including other scales not used for this study, for capturing the semantic attributes of student responses during vocabulary learning.

Also, there were some limitations in the current study and areas for future work. First, expanding the scope of analysis to the full set of experimental conditions used in the study may reveal more complex interactions between these conditions and students' short- and long-term learning. Second, this study used a fixed threshold of 0.5 for investigating false prediction results. However, an optimal threshold for each participant group or prediction model could be selected, especially if there are different false positive or negative patterns observed for different groups of students. Lastly, this study collected data from a single vocabulary tutoring system that was used in a classroom setting. Applying the proposed method to data that was collected from a non-classroom setting or other vocabulary learning system would be useful to show the generalization of our suggested method.

6. ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A140647 to the University of Michigan. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education. We thank Dr. Charles Perfetti and his lab team at the University of Pittsburgh, particularly Adeetee Bhide and Kim Muth, and the helpful personnel at all of our partner schools.

7. REFERENCES

[1] S. Adlof, G. Frishkoff, J. Dandy, and C. Perfetti. Effects of induced orthographic and semantic knowledge on subsequent learning: A test of the partial knowledge hypothesis. *Reading and Writing*, 29(3):475–500, 2016.

[2] J. E. Beck. Engagement tracing: Using response times to model student disengagement. *Artificial Intelligence in Education: Supporting Learning Through Intelligent and Socially Informed Technology*, 125:88, 2005.

[3] K. Collins-Thompson and J. Callan. Automatic and human scoring of word definition responses. In *HLT-NAACL*, pages 476–483, 2007.

[4] M. Coltheart. The MRC psycholinguistic database. *The Quarterly Journal of Experimental Psychology*, 33(4):497–505, 1981.

[5] E. Dale. Vocabulary measurement: Techniques and major findings. *Elementary English*, 42(8):895–948, 1965.

[6] F. T. Durso and W. J. Shore. Partial knowledge of word meanings. *Journal of Experimental Psychology: General*, 120(2):190, 1991.

[7] G. A. Frishkoff, K. Collins-Thompson, L. Hodges, and S. Crossley. Accuracy feedback improves word learning from context: Evidence from a meaning-generation task. *Reading and Writing*, 29(4):609–632, 2016.

[8] G. A. Frishkoff, K. Collins-Thompson, S. Nam, L. Hodges, and S. A. Crossley. Dynamic support of contextual vocabulary acquisition for reading (DSCoVAR): An intelligent tutoring system for contextual word learning. *Handbook on Educational Technologies for Literacy*, 2016.

[9] G. A. Frishkoff, C. A. Perfetti, and K. Collins-Thompson. Predicting robust vocabulary growth from measures of incremental learning. *Scientific Studies of Reading*, 15(1):71–91, 2011.

[10] T. K. Landauer. *Latent Semantic Analysis*. Wiley Online Library, 2006.

[11] Y. Li, L. Xu, F. Tian, L. Jiang, X. Zhong, and E. Chen. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina*, pages 3650–3656, 2015.

[12] Y. Ma, L. Agnihotri, M. H. Education, R. Baker, and S. Mojarad. Effect of student ability and question difficulty on duration. In *Educational Data Mining*, 2016.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[14] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[15] S. Nam. Predicting off-task behaviors in an adaptive vocabulary learning system. In *Educational Data Mining*, 2016.

[16] C. E. Osgood, G. J. Suci, and P. H. Tannenbaum. *The Measurement of Meaning*. University of Illinois Press, 1957.

[17] K. Ostrow, C. Donnelly, S. Adjei, and N. Heffernan. Improving student modeling through partial credit and problem difficulty. In *Proc. of the Second ACM Conference on Learning@Scale*, pages 11–20. ACM, 2015.

[18] P. I. Pavlik and J. R. Anderson. Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cog. Science*, 29(4):559–586, 2005.

[19] E. G. Van Inwegen, S. A. Adjei, Y. Wang, and N. T. Heffernan. Using partial credit and response history to model user knowledge. *International Educational Data Mining Society*, 2015.

[20] L. M. Yonek. *The Effects of Rich Vocabulary Instruction on Students' Expository Writing*. PhD thesis, University of Pittsburgh, 2008.

Generalizability of Face-Based Mind Wandering Detection Across Task Contexts

Angela Stewart
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN, 46556, USA
astewa12@nd.edu

Nigel Bosch
University of Illinois at Urbana-Champaign
1205 West Clark Street
Urbana, IL, 61801, USA
pnb@illinois.edu

Sidney K. D'Mello
University of Notre Dame
118 Haggar Hall
Notre Dame, IN, 46556
sdmello@nd.edu

ABSTRACT

We investigate generalizability of face-based detectors of mind wandering across task contexts. We leveraged data from two lab studies: one where 152 college students read a scientific text and another where 109 college students watched a narrative film. We automatically extracted facial expressions and body motion features, which were used to train supervised machine learning models on each dataset, as well as a concatenated dataset. We applied models from each task context (scientific text or narrative film) to the alternate context to study generalizability. We found that models trained on the narrative film dataset generalized to the scientific text dataset with no modifications, but the predicted mind wandering rate needed to be adjusted before models trained on the scientific text dataset would generalize to the narrative film dataset. Additionally, we analyzed generalizability of individual features and found that the lip tightener and jaw drop action units had the greatest potential to generalize across task contexts. We discuss findings and applications of our work to attention-aware learning technologies.

Keywords

Mind Wandering, Mental States, Attention Aware Interfaces, Cross-Corpus training.

1. INTRODUCTION

Consider a typical day when you were an undergraduate college student. Your first class is your favorite, so you are engaged in the lecture content and processing new information. In your next class, you watch a documentary about a subject that does not interest you, causing your attention to focus on unrelated thoughts of your social life, rather than processing the information in the video. Later, you work on a homework assignment that you find frustrating, leading to waning motivation. Towards the end of your day, you attend a chemistry lab, where you interact with a new educational game that teaches you the basics of chemical bonds. At some points you are enjoying the game, and thus engaged in deeply learning the content. However, you later become bored during a long period of repetitive gameplay, causing you to become distracted and miss important information. Throughout the day, your mental states (engagement, frustration, boredom) influenced your learning. Your learning experience could have been augmented with technology that responded to your changing mental state, thus assisting you in achieving the most effective learning experience.

Educational interfaces that detect and respond to student mental states are driven by work on cognitive and affective state modeling, which has been investigated for many years. For example, attention and affect has been modeled in educational tasks such as reading comprehension [6, 16, 28] and computerized tutoring [3, 19], among others. In general, there has been a plethora of work that has modeled a variety of mental states within specific educational tasks (e.g., [2, 15, 19]) to better understand these states and use that knowledge to facilitate student learning.

However, prior research has overwhelmingly investigated single task contexts, and has overlooked generalizability to different contexts. For example, models that track attention during reading might not generalize to lecture viewing, educational gaming, and so on. This makes it difficult to decouple task-specific effects from more fundamental patterns. In contrast, models that successfully generalize across multiple contexts should reveal observable signals (i.e. eye gaze, facial features, and physiology data) that are general, rather than task-specific. Models using such indicators will be key to developing adaptive technologies that are sensitive to student mental states and that can operate across a range of educational activities.

We report results on modeling mental states in a generalized way using mind wandering (MW) as a case study. MW is a ubiquitous phenomenon where thoughts shift from task-related processing to task-unrelated thoughts [15]. MW is estimated to occur anywhere from 20% - 50% of the time, depending on the person, task, and environmental context [23]. It is has also been associated with lower performance on a variety of educational tasks, such as reading comprehension [16] and retention of lecture content [29], thus impacting student learning.

As with work on other mental states, research on MW has largely failed to address models that generalize across contexts [6, 15]. MW detection has been investigated in reading comprehension [6, 16], narrative and instructional film comprehension [25, 26], and student interaction with an intelligent tutoring system (ITS) [19]. To our knowledge, no work has investigated MW detection with the goal of generalizability across task contexts.

We specifically investigate the generalizability of MW models across two task contexts - reading a scientific text and viewing a narrative film. These contexts were chosen because of their broad applicability to education in the classroom and online. For example, a documentary film could be shown in a sociology course or distance learning students could read instructional texts prior to engaging in an online discussion.

1.1 Related Work

Cross corpus training has been researched in a variety of classification problems, such as sentiment analysis [31] and acoustic-based emotion recognition [35]. Cross corpus training seeks to improve robustness of machine-learned models by leveraging multiple datasets in classifier training and testing. For example, Webb and Ferguson [32] applied cross corpus training techniques to characterize the function of segments of dialogue using automatically extracted lexical and syntactic features called cue phrases. Each extracted cue phrase was used to classify a segment of dialogue. They trained separate classifiers on two different datasets, and applied the classifier to the dataset on which it was not trained. They found the cross-training results were comparable to the results of training and testing on the same dataset (e.g. the best cross-trained classifier achieved and accuracy of 71%, compared to an accuracy of 81% when trained and tested on the same dataset). Additionally, they examined generalizability of the cue phrases across datasets by reducing the feature set to contain only cues present in both datasets. They found that reducing the feature set yielded slight improvements, and demonstrated the discriminative nature of a small number of features.

Zhang et. al. [35] similarly explored the use of multiple datasets for creating context-generalizable models. They built classifiers for valence and arousal on highly varied emotional speech datasets using a leave-one-corpora-out cross-validation technique. Additionally, they explored methods for data normalization (within each dataset and between datasets) and agglomeration of both labeled and unlabeled data. They found that, of their six emotional speech corpora, training on some subsets yielded higher accuracy than others. Their work suggested that careful selection of corpora best suited for training might yield better emotional speech recognition performance than an all-or-nothing approach to cross-corpus training.

Our work approaches cross-corpus modeling through detection of MW. A variety of studies have investigated MW detection during educational tasks, such a reading [15], interacting with an intelligent tutoring system (ITS) [19], or watching an educational video [26]. No work has focused on MW from a cross-corpus modeling perspective, to our knowledge, so we review the individual studies below.

Detection of MW from eye gaze features while reading has been amply investigated. For example, Bixler and D'Mello [4] built models to detect MW while students read texts about scientific research methods. This work made use of probe-caught reports (students respond yes or no to auditory thought probes of whether they were MW), instead of self-caught reports (students report whenever they catch themselves MW). Their analysis of eye gaze features showed that certain types of fixations were longer during MW. Specifically, they found that longer gaze fixations (consecutive fixations on a single word), first-pass fixations (fixations on a word during the first pass through a text), and single fixations (fixations on a word only fixated on once) were predictive of MW. In other work, Bixler and D'Mello [5] similarly used eye gaze features, but used self-caught reports of MW. They found that a greater number of fixations, longer saccade length, and line cross saccades were indicative of MW. Across studies on MW detection during reading, longer fixations were found to be indicative of MW [4, 15, 28], suggesting these features might generalize well.

Pham and Wang [26] similarly used consumer-grade equipment to detect MW while students watched videos from massively open online courses (MOOCs). They made use of heart rate, detected by monitoring fingertip blood flow, using the back camera of a smartphone (i.e., photoplethysmography). Their models achieved a 22% improvement over chance. Although their method for detecting MW could be implemented across a variety of tasks, the question of whether heart rate is indicative of MW across task contexts has not yet been investigated.

Hutt et. al. provided limited evidence of generalizability of MW detection across different learning tasks during student interaction with an ITS [19]. They employed a genetic algorithm to train a neural network using context-independent eye-gaze features and context-dependent interaction features (e.g., current progress within the ITS). They achieved an F_1 value of .490 (chance = .190). This work provided some evidence of generalizability because the visual stimuli and interaction patterns varied throughout. For example, students interacted with an animated pedagogical agent in a scaffolded dialogue phase and completed concept maps without the tutoring agent in another interaction phase. However, it is still unclear if their model would generalize to a broader range of tasks, particularly less interactive ones like reading or film viewing. Furthermore, their best-performing models used context-dependent features, which could prevent the detector from generalizing to a task where those features could not be used.

1.2 Novelty

Our contribution is novel in a variety of ways. First, we demonstrate the feasibility of building cross-context detectors of mental states, specifically MW. Further, previous work on MW detection has sometimes made use of context-specific features (e.g., reading times) that are not expected to generalize to other contexts [19, 25]. In contrast, our work detects MW using only facial features and upper body movement, recorded using commercial-off-the-shelf (COTS) webcams that are expected to generalize more broadly. Additionally, the use of COTS webcams support a broader implementation of MW detectors as webcams are ubiquitous in modern technology. This is in contrast to prior research that has used specialized equipment, like eye trackers [15, 19, 25] or physiology sensors [7], which students would likely not have access to.

2. DATASETS

This study makes use of narrative film [23] and scientific reading comprehension [22] datasets collected as part of a larger project. Here, we include details pertaining to video-based detection of MW.

2.1 Narrative Film Comprehension

Participants were 68 undergraduate students from a medium-sized private Midwestern university and 41 undergraduate students from a large public university in the Southern United States. Of the 109 students, 66% were female and their average age was 20.1 years. Students were compensated with course credit. Data from four students were discarded due to equipment failure.

Students viewed the narrative film *The Red Balloon* (1956), a 32.5-minute French-language film with English subtitles (Figure 1). The film has a musical score but only sparse dialogue. This short fantasy film depicts the story of a young Parisian boy who finds a red helium balloon and quickly discovers it has a mind of its own as it follows him wherever he goes. This film was selected because of the low likelihood that participants have previously seen it and because it has been used in other film comprehension studies [34].

measurement necessitate, but which in more enlightened countries are wholly unnecessary. This book is not prepared to meet the requirements and artificial restrictions of any syllabus, and it is not prepared to help students through any examination. I cannot help thinking, however, that if the type of student who puts more faith in learning formulae

Figure 1. A screenshot of the narrative film (left) and scientific text (right) are shown.

Students' faces and upper bodies were recorded with a low-cost ($30) consumer-grade webcam (Logitech C270).

Students were instructed to report MW throughout the film by pressing labeled keys on the keyboard. Specifically, students were asked to report a task-unrelated thought if they were "thinking about anything else besides the movie" and a task-related interference if they were "thinking about the task itself but not the actual content of the movie." A small beep sounded to register their report, but film play was not paused. After viewing the film, students took a short test about the content and completed additional measures not discussed further.

We recorded a total of 1,368 MW reports from the 105 participants with valid video recordings. In this work, we do not distinguish between the two types of MW, instead merging the task-unrelated thoughts and the task-related interferences, both of which represent thoughts independent of the content of the film.

2.2 Scientific Reading Comprehension

Participants were 104 undergraduate students from a medium-sized private Midwestern university and 48 undergraduate students from a large public university in the Southern United States. Of the 152 participants, 61% were female and their average age was 20.1 years. Participants were compensated with course credit. Data from eight participants were discarded due to equipment failure.

Students read an excerpt from *Soap-Bubbles and the Forces which Mould Them* [8]. Like *The Red Balloon* (Figure 1), we chose this text because its content would likely be unfamiliar to a majority of readers. The text contained around 6,500 words from the first chapter of the book. In all, 57 pages (screens of text) with an average of 115 words each were displayed on a computer screen in 36-pt Courier New typeface. The only modification to the text was the removal of images and references to them after verifying that these were not needed for comprehension.

Students who read the scientific text were instructed to report MW in the same way as those who watched the narrative film. They were instructed to report a task-unrelated thought if they were "thinking about anything else besides the task" and a task-related interference if they were "thinking about the task itself but not the actual content of the text." Participants completed a comprehension assessment after reading the text. We recorded a total of 3,168 MW reports from the 144 students with valid video recordings.

2.3 Self Reports of MW

MW was measured via self-reports in both studies, so it is prudent to discuss the validity of self-reports. We used self-reports because this is currently the most common approach to measure an inherently internal (but conscious) phenomenon [5, 15]. Self-reported MW has been linked to predictable patterns in physiology [30], pupillometry [17], eye-gaze [28] and task performance [27], providing evidence for the convergent and predictive validity for this approach. To improve the quality of self-reports, we encouraged students to report honestly and assured them that reporting MW would not in any way effect the credit they received for participation.

The alternative to using self-caught reports is using probe-caught reports, which require a student response to a thought-probe (e.g., a beep). We chose self-caught reports over the probe-caught because the probe-caught method can potentially interrupt the comprehension process (i.e., when participants report "no" to the probes). Interruptions are particularly problematic in the film comprehension task, as participants did not have control over the media presentation (i.e., no pausing or rewinding of the film). Furthermore, it is also unclear if a probe-caught report takes place at the beginning or end of MW, or somewhere in between. Conversely, self-caught reports are likely to occur at the end of a MW episode when the student became aware that they were not attending to the task at hand.

3. MACHINE LEARNING

We explored a variety of machine learning techniques for cross-context MW detection using the same approach to segmenting instances and constructing features for both datasets.

3.1 Segmenting Instances

Reports of MW were distributed throughout the course of the film viewing or text reading session. We created instances that corresponded to reports of MW by first adding a 4-second offset prior to the report. This was done to ensure that we captured participants' faces while MW vs. in the act of reporting MW itself (i.e., the preparation and execution of the key press). This 4-second offset was chosen based on four raters judgements of whether or not movement related to the key-press could be seen within offsets ranging from 0 to 6 seconds. Data was then extracted from the 20 seconds prior to the MW report. A window size of 20 seconds was chosen based on prior experimentation that sought to balance creating as many instances as possible (shorter window sizes) and having sufficient data in each window (longer window sizes) to detect MW.

We extracted "not MW" instances from windows of data between MW reports. The entire session (reading or video watching) was divided into 24-second segments (20 second windows of data and a 4 second offset as with the MW segments). Any segments

overlapping the 30 seconds prior to a MW report were discarded. We do not know precisely when MW starts, so we chose to discard instances overlapping the 30 seconds prior to MW reports, to separate students when they were actually MW from when they were not. We also discarded any segments overlapping a page turn (discussed in Section 3.2). All remaining segments were labeled Not MW. Our approach to segmenting instances is shown in Figure 2.

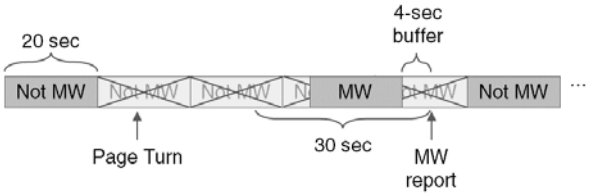


Figure 2. Illustration of the instance extraction method.

3.2 Instance Selection

A full accounting of the instance selection process is shown in Table 1. Our goal was to make the two data sets as similar as possible so that task-specific effects could be studied without additional confounds.

We first discarded any instances where there was less than one second of usable data in that time window. Data was not usable when the student's face was occluded due to extreme head pose or position, hand-to-face gestures, and rapid movements. Additionally, for the scientific reading dataset, we discarded instances that overlapped with page turn events. In prior experimentation, we trained a model to detect MW using only a binary feature of whether or not that instance overlapped a page turn boundary. MW was detected at rates above chance in this experimental model. Therefore, we concluded that including instances that overlapped page turn boundaries would inflate performance as the detector could simply be picking up on the act of pressing the key to advance to the next page.

After discarding instances using the method above, we matched the scientific reading and narrative film datasets on school (medium-sized Midwestern private university or large Southern public university), reported ethnicity, and reported gender. The scientific reading dataset was randomly downsampled to contain approximately the same number of students in each gender, race, or school category, as the film dataset. This participant-level matching on school, ethnicity, and gender was done to eliminate external sources of variance that could influence MW detection, potentially obfuscating task effects from population effects.

Finally, the datasets were downsampled to contain equal numbers of instances because the size of the training set is known to bias classifier performance [13]. We also downsampled the data to achieve a 25% MW rate in order to be consistent with research that suggests that MW occurs between 20% and 30% of the time during reading and film comprehension [6, 23]. Further, the MW rates of 30% and 14% obtained in these data are more artefacts of the instance segmentation approach rather than the objective rate, so resampling ensures a dataset that is more reflective of expected MW rates.

Table 1. An accounting of instance selection process

	Reading (% MW)	Film (% MW)
Base	7,267 (30%)	7,313 (14%)
Face Detected	7,266 (30%)	7,238 (14%)
Page Boundary	1,400 (36%)	N/A
Participant Matching	1,273 (35%)	N/A
Downsampling	1,100 (25%)	1,100 (25%)

3.3 Feature Extraction and Selection

We used commercial software, the Emotient SDK [36] to extract facial features. The Emotient SDK, a version of the CERT computer vision software [24] (Figure 3) provides likelihood estimates of the presence of 20 facial action units (AUs; specifically 1, 2, 4, 5, 6 ,7, 9, 10, 12, 14, 15, 17, 18, 20, 23, 24, 25, 26, 28, and 43 [14]) as well as head pose (orientation), face position (horizontal and vertical within the frame), and face size (a proxy for distance to camera). Additionally, we used a validated motion estimation algorithm to compute gross body movements [33]. Body movement was calculated by measuring the proportion of pixels in each video frame that differed by a threshold from a continuously updated estimate of the background image generated from the four previous frames.

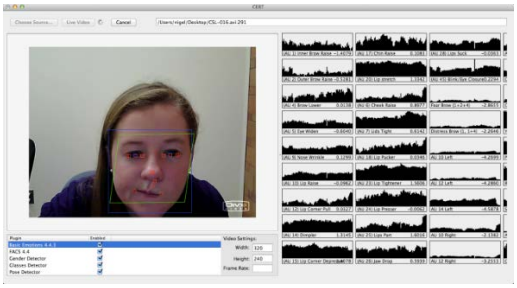


Figure 3. Interface demonstrating AU estimates detected from a face video.

Features were created by aggregating Emotient estimates in a window of time leading up to each MW or Not MW instance using minimum, maximum, median, mean, range, and standard deviation for aggregation. In all, there were 162 facial features (6 aggregation functions × [20 AUs + 3 head pose orientation axes + 2 face position coordinates + face size + Motion]). Outliers (values greater than three standard deviations from the mean) were replaced by the closest non-outlier value in a process called Winsorization [11].

We used tolerance analysis to eliminate features with high multicollinearity (variance inflation factor > 5) [1], after which, 37 features remained. This was followed by RELIEF-F [21] feature selection (on the training data only) to rank features. We retained a proportion of the highest ranked features for use in the models (proportions ranging from .05 to 1.0 were tested). Feature selection was performed using nested cross-validation on training data only. We ran 5 iterations of feature selection within each cross-validation fold (discussed below), using data from a randomly chosen 67% of students within the training set in each iteration.

3.4 Supervised Classification and Validation

Informed by preliminary experiments, we selected seven classifiers for more extensive tests (Naïve Bayes, Simple Logistic Regression, LogitBoost, Random Forest, C4.5, Stochastic Gradient Descent, and Classification via Regression) using the WEKA data mining

toolkit [18]. For each classifier, we applied SMOTE [9] to the training set only. SMOTE, a common machine learning technique for dealing with data imbalance, creates synthetic interpolated instances of the minority class to increase classification performance.

We evaluated the performance of our classifiers using leave-one-participant-out cross-validation. This process runs multiple iterations of each classifier in which, for each fold, the instances pertaining to a single participant are added to the test set and the training set is comprised of the instances for the other participants. Feature selection was performed on a subset of participants in the training set. The leave-one-out process was repeated for each participant, and the classifications of all folds were weighted equally to produce the overall result. This cross-validation approach ensured that in each fold, data from the same participant was in the training set or testing set but never both, thereby improving generalization to new participants.

Accuracy (recognition rate) is a common measure to evaluate performance in machine learning tasks. However, any classifier that defaults to predicting the majority class label of an imbalanced dataset can appear to have high accuracy despite incorrect predictions of all instances of the minority class label [20]. This is particularly detrimental in applications where detecting the minority class is of upmost importance. In our task, we prioritized the detection of MW despite the large imbalance in our dataset. Therefore, we considered the F_1 score for the MW label as our key measure of detection accuracy since F_1 attempts to strike a balance between precision and recall.

4. RESULTS

4.1 Cross-dataset Training and Testing

We trained three classifiers: one on the scientific text dataset, one on the narrative film dataset, and one on a concatenated dataset comprised of the first two. For each of the three training sets, the classifier that yielded the highest MW F_1 is shown in Table 2. We used leave-one-student-out cross validation for within-dataset evaluations. Conversely, to measure generalizability of the models across contexts we applied the classifier trained on scientific text data to the narrative film data, and vice versa. We compared our model to a chance model that classified a random 25% (MW prior proportion) of the instances as MW. This chance-level method yielded a precision and recall of .250 (equal to the MW base rate).

Table 2. Results for the models with highest MW F_1 for the within-data set validation (cross-training results in parentheses).

Training Set	Classifier	MW F_1	Precision	Recall
Scientific Text	Logitboost	.441 (.267)	.376 (.252)	.553 (.284)
Narrative Film	C4.5	.436 (.407)	.303 (.278)	.775 (.760)
Both	Logistic	.424	.314	.655

We calculated improvement over chance as (actual performance – chance)/(perfect performance – chance). All three models showed improvement over chance (25% for scientific text, 25% for narrative film, and 23% for the concatenated dataset) when trained and tested on the same dataset. When tested on the alternative dataset, the narrative film classifier generalized well to the scientific text dataset (21% improvement over chance). However, the scientific text model showed chance-level performance on the narrative film corpus (2% improvement over chance). The MW F_1 of the concatenated dataset model was simply an average of the MW F_1 score of the individual datasets when the instance predictions of the individual datasets are separated (.413 for the scientific reading dataset and .436 on the narrative film dataset). These results showed that the concatenated classifier does not skew towards predicting one dataset better than the other, but rather predicts both models with comparable accuracy.

Table 2 also shows precision and recall for each of the models. Across all models, recall was higher than precision, indicating a lot false positives. It is important to note the near chance-level recall and precision of the model trained on scientific reading data when applied to the narrative film data. The lack of improvement over chance for both recall and precision demonstrated the need to improve generalizability in both dimensions. Conversely, the cross-trained narrative film model had lower precision, but good recall, resulting in an improved MW F_1 score.

4.2 Classifier Generalizability

To address the negligible improvement over chance of the scientific text model when tested on the narrative film dataset, we repeated the training and testing using C4.5 as the classifier. The C4.5 classifier was chosen because it generalized better when trained on the narrative film dataset than the Logitboost classifier generalized when trained on the scientific text dataset. The results are shown in Table 3, where we note no notable improvement over the previous Logitboost classifier in Table 2 (change from .267 to .287 when tested on the narrative film dataset). Therefore, the lack of evidence for generalizability for the scientific text model could be due to overfitting to the training set, rather than classifier selection.

Table 3. Results (MW F_1) for the C4.5 classifier for within- and cross- validation.

Training Set	Within	Cross
Scientific Text	0.425	0.287
Narrative Film	0.436	0.407
Both	0.415	N/A

4.3 Prediction Threshold Adjustment

We further investigated the lack of generalizability of the scientific text model by considering the MW prediction rate. We compared the performance of both models on the narrative film dataset. Recall dropped considerably more than precision (Table 2; recall dropped from .775 to .284; precision decreased from .303 to .252). We hypothesized that recall decreased because of a difference in predicted MW rates (Table 4). In fact, the predicted MW rate in the narrative film data dropped from 64% to 28% when applying the scientific text model to the same data. This supported our hypothesis that the low recall was linked to lower predicted MW rates. Furthermore, 39% of the correctly classified instances (true positives and true negatives) were MW when applying the narrative film model to the narrative film data compared to 12% for the scientific text model applied to the same data. This demonstrated that the scientific text model was much more prone to missing MW instances, further supporting our hypothesis.

To address this, we adjusted the predicted MW rate of the scientific text model when applied to the narrative film dataset. The classifier outputs a likelihood of MW and we previously considered instances with likelihoods greater than .5 as MW. We adjusted that prediction threshold from .1 to 1 in increments of .1 (Figure 4) to investigate how changes in predicted MW rate (higher for lower thresholds) effected recall, and thus MW F_1.

Table 4. Predicted MW Rates.

Training Set	Within	Cross
Scientific Text	38%	**28%**
Narrative Film	**64%**	68%
Both	52%	N/A

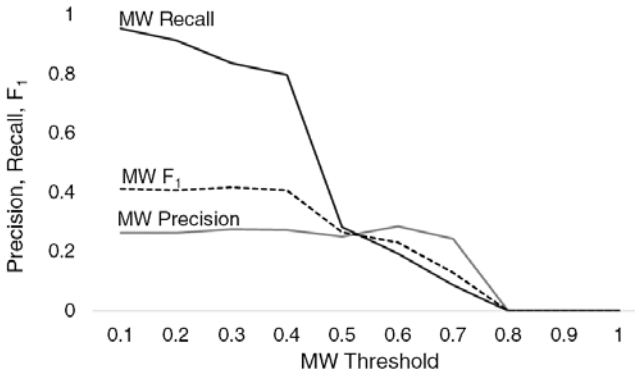


Figure 4. MW precision, recall, and F_1 as the prediction threshold varies for the scientific text model applied to the narrative film dataset.

We note that MW F_1 score degrades at a threshold of .5. We adjusted the threshold to .3 and yielded the results shown in Table 5. After adjusting the MW prediction threshold, both precision and recall of the narrative film data applied to the scientific text model showed comparable performance to the cross-trained narrative film model. It is important to note that the adjusted MW prediction threshold yielded a predicted MW rate of 76%, much higher than the MW rate of the dataset (25%). As with the generalized narrative film model, this reduced precision because the high predicted MW rate produced a large number of false positives.

Table 5. Results for models with highest MW F_1 (cross-training results in parentheses). Cross-training results for the scientific text model reflect a MW prediction threshold of .3.

Training Set	Classifier	MW F_1	Precision	Recall
Scientific Text	Logitboost	.441 (**.416**)	.376 (**.276**)	.553 (**.836**)
Narrative Film	C4.5	.436 (.407)	.303 (.278)	.775 (.760)
Both	Logistic	.424	.314	.655

4.4 Feature Analysis

We analyzed the facial features to further study generalizability by predicting MW with different subsets of the entire feature set. The C4.5 classifier was chosen for this feature analysis because of its consistency on both the scientific text model and concatenated dataset. Each subset consisted of the features (e.g., median, standard deviation) from one AU, or from face position, size, orientation, or motion. Since tolerance analysis was not used here, we only considered the minimum, maximum, median, and standard deviation aggregated features to prevent redundancy (e.g., between median and mean). For example, we used the minimum, maximum, median, and standard deviation feature values for AU5 (upper lid raiser) to predict MW. This approach was applied to the 20 AU subsets, as well as face position, size, orientation, and motion subsets. We generated the same cross-training configurations of in Section 4.1 (i.e., train on scientific text, test on narrative film, etc.).

To rank the subsets of features on generalizability, we examined MW F_1 scores when testing on the alternative dataset only. For example, using the AU9 (nose wrinkle) subset, we investigated MW F_1 value of scientific text model applied to the narrative film dataset and the narrative film model applied to the scientific text dataset. Table 4 shows these results only for features that achieved a MW F_1 of greater than .250 (chance) on all dimensions (within dataset validation and cross-training). We selected features for further analysis if their MW F_1 was greater than .300 for both cross-training results. This value of .300 was used to filter out features that performed well on the within-dataset validation, but fell short on cross training. It also ensured that a feature performed better than chance on both cross-trained results (i.e., train on narrative film and test on scientific text, and vice versa), rather than only generalizing to one dataset. Using this criterion, only AU23 and AU26 showed notable improvement over chance.

We used the C4.5 classifier to generate the same models in Table 2 (train/test scientific text, train scientific text/test narrative film, etc.) using only the features from AU23 and AU26 (Table 7). None of these models (scientific text, narrative film, or concatenated) achieved a MW F_1 as high as those in Table 2, which used a combination of tolerance analysis and RELIEF-F to select features. This suggested that, while AU23 and AU26 might individually predict MW, when used together, their prediction power might be limited, compared to other feature selection techniques.

Table 6. MW F_1 score for within-data set validation with cross-data set scores (in parentheses).

	Training Set	
Facial Feature	**Scientific Text**	**Narrative Film**
AU4 (brow lowerer)	.378 (.278)	.398 (.395)
AU6 (cheek raiser)	.369 (.259)	.361 (.321)
AU9 (nose wrinkler)	.300 (.268)	.392 (.303)
AU14 (dimpler)	.303 (.267)	.383 (.376)
AU23 (lip tightener)	**.334 (.333)**	**.363 (.317)**
AU26 (jaw drop)	**.414 (.321)**	**.365 (.357)**
Face Height (size)	.322 (.256)	.339 (.289)
Face X (position)	.404 (.316)	.382 (.282)

Table 7. Results for models when only using the C4.5 classifier on AU23 and AU26.

Training Set	Classifier	MW F_1	Precision	Recall
Scientific Text	C4.5	.383 (.272)	.255 (.206)	.764 (.404)
Narrative Film	C4.5	.397 (.257)	.333 (.235)	.491 (.284)
Both	C4.5	.368	.271	.575

5. ANALYSIS

We developed automated detectors of MW using video-based features in the contexts of narrative film viewing and scientific reading. The generalizability of these models was dependent on corpora on which the model was trained and the rate at which the model predicts MW. In this section, we discuss our main findings and applications of this work. We also discuss limitations and future work.

5.1 Main Findings

We expanded on previous MW detection work through cross-context modeling. We trained three models on three datasets

(scientific text, narrative film, and a dataset concatenated from the two). We found each of these models (trained and tested on the same corpus) performed at a notable 23% to 25% improvement over chance. This demonstrated the feasibility of detecting MW on individual corpora. However, recall was greater than precision, indicating prediction of false positives. This should be considered when implementing MW detectors in educational environments where excessive prediction of student MW could be demotivating.

We investigated generalizability of the single-dataset models (i.e. scientific text or narrative film) by applying the model to the dataset on which it was not trained. The model trained on the narrative film dataset maintained performance when applied to the scientific text dataset (Table 2), providing some evidence for generalizability, but this performance was boosted by high recall (and comparatively low precision). Precision and recall (and thus MW F_1) were near chance-level when the model trained on the scientific text dataset was applied to the narrative film dataset, suggesting that the model might overfit to the scientific text training set.

We attempted to address this problem by applying the C4.5 classifier, as it comparatively generalized well when trained on the narrative film dataset. MW F_1 score for the scientific text classifier applied to the narrative film data again negligibly increased. This suggested that the training data (only scientific text) used was not appropriate for model generalization. This idea is supported by the performance of the narrative film model on the scientific text data (although detection of false positives is a limitation) and the notable improvement over chance (22% to 23%) for the concatenated dataset. The performance of both models suggested that there were discernable similarities between MW instances across the two datasets, which can be detected using our techniques.

In addition to training data, we also found that predicted MW rate effected model generalizability. We adjusted MW predictions according to a sliding threshold for the narrative film predictions obtained from the scientific text model. We found that relaxing the criteria for classifying an instance as MW (i.e. adjusting the likelihood prediction threshold from .5 to .3) yielded results comparable to the cross-trained narrative film model. However, this approach to increasing recall should be used with caution as it leads to increased likelihood of false positives. Perhaps in a real-time MW intervention scenario, a more balanced approach could be taken where the MW likelihood prediction is used to determine if a MW intervention is triggered (e.g., if the detector determines there is a 40% likelihood the student is MW, then there is a 40% chance a MW intervention is triggered).

We detected MW using individual feature subsets to ascertain whether certain face-based features (i.e. AUs, head orientation, position, size, and motion) generalize. We found two feature subsets (AU23 – lip tightener and AU26 – jaw drop) that showed a MW F_1 of at least .300 on both cross-trained models. It is notable that when looking at the generalizability of these features, they did not individually achieve MW F_1 scores as high as the best performing models in Table 2. This demonstrated the need for multiple features to work together to detect MW, rather than relying on a single feature. Furthermore, this showed that our method of feature selection (tolerance analysis and selecting a proportion of features using RELIEFF) was important to model performance.

5.2 Applications

The present findings are applicable to educational user interfaces that involve reading or film comprehension. Monitoring and responding to MW could greatly improve student performance on these tasks. Films and instructional texts play a major role in learning (both in the classroom and online). For example, films can give historical background on a time period being discussed in literature classes and instructional texts can supplement lecture content through textbooks or technical articles. Due to the relationship between MW and low task performance, user interfaces that detect and respond to MW in contexts where attention is key (i.e. education) would help students remain focused on their learning.

These findings are particularly promising for implementation in massively open online courses (MOOCs). Our method for detecting MW exclusively uses COTS webcams. These webcams are ubiquitous in today's computers and mobile devices; thus our work would integrate into a variety of learning environments without extra cost. Such a video-based detector of MW could feasibly respond to student MW through suggesting a student revisit text or video content, asking a reengaging question, or advising the student to take a break.

5.3 Limitations and Future Work

While we demonstrated techniques for modeling generalizability across task contexts, our work has a few limitations. First, precision is moderate, even on our best models. High predicted MW rates lead to high recall, but also more false positives. In this work, we chose to accept this tradeoff, with the goal of generalizability in mind. However, raising precision, while maintaining recall is key to task-generalizable MW detectors being successful in educational environments. Since MW is the minority class (25% of all instances), investigating skew-insensitive classifiers, such as Hellinger Distance Decision Trees [10], could improve precision.

Additionally, this work focuses exclusively on generalizability from the perspective of task context (viewing a narrative film vs. reading a scientific text). Claims of generalizability could be strengthened through MW detection across environments. Both the narrative film and scientific reading datasets were collected in a controlled lab setting. MW detection in the field, such as computer-enabled classrooms or the personal workstations of MOOC users, should be considered prior to implementation in such environments. Furthermore, student generalizability should be further examined. In this work, we detect MW in a student-independent way. However, participants were all of similar age and enrolled in college. Future work could examine the generalizability of our method for detecting MW in non-college-aged students, such as elementary students in a computer-enabled classroom or non-traditional students enrolled in distance learning courses.

5.4 Concluding Remarks

In this work, we showed evidence that generalizable detectors of MW can be created using video-based features. The corpora used to train models of MW and predicted MW rates both play a role in the model's ability to generalize and should be considered as work on cross-context MW generalization advances. This work advances the field of attention-aware interfaces [12] by demonstrating the feasibility of modeling MW across the educational contexts of reading a scientific text and viewing a narrative film. Our approach to detecting MW is the first step towards building interfaces that detect MW across multiple educational activities.

6. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSF) (DRL 1235958 and IIS 1523091). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

7. REFERENCES

[1] Allison, P.D. 1999. *Multiple regression: A primer*. Pine Forge Press.

[2] Baker, R.S. et al. 2012. Towards automatically detecting whether student learning is shallow. *International Conference on Intelligent Tutoring Systems* (Chania, Crete, Greece, 2012), 444–453.

[3] Baker, R.S. et al. 2012. Towards sensor-free affect detection in a Cognitive Tutor for Algebra. *Educational Data Mining* (Chania, Crete, Greece, 2012).

[4] Bixler, R. and D'Mello, S. 2016. Automatic gaze-based user-independent detection of mind wandering during computerized reading. *User Modeling and User-Adapted Interaction*. 26, 1 (2016), 33–68.

[5] Bixler, R. and D'Mello, S.K. 2015. Automatic Gaze-Based Detection of Mind Wandering with Metacognitive Awareness. *User Modeling, Adaptation and Personalization: 23rd International Conference* (Dublin, Ireland, 2015), 31–43.

[6] Bixler, R. and D'Mello, S.K. 2014. Toward fully automated person-independent detection of mind wandering. *Proceedings of the 22nd International Conference on User Modeling, Adaptation, and Personalization* (Switzerland, 2014), 37–48.

[7] Blanchard, N. et al. 2014. Automated physiological-based detection of mind wandering during learning. *Intelligent Tutoring Systems* (Honolulu, Hawaii, USA, 2014), 55–60.

[8] Boys, C.V. and others 1890. *Soap-bubbles, and the forces which mould them*. Cornell University Library.

[9] Chawla, N.V. et al. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*. (2002), 321–357.

[10] Cieslak, D.A. et al. 2012. Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*. 24, 1 (2012), 136–158.

[11] Dixon, W.J. and Yuen, K.K. 1974. Trimming and winsorization: A review. *Statistische Hefte*. 15, 2–3 (1974), 157–170.

[12] D'Mello, S.K. 2016. Giving Eyesight to the Blind: Towards Attention-Aware AIED. *International Journal of Artificial Intelligence in Education*. 26, (2016), 645–659.

[13] Domingos, P. 2012. A few useful things to know about machine learning. *Communications of the ACM*. 55, 10 (2012), 78–87.

[14] Ekman, P. and Friesen, W.V. 1977. *Facial action coding system*.

[15] Faber, M. et al. 2017. An automated behavioral measure of mind wandering during computerized reading. *Behavior Research Methods*. (2017), 1–17.

[16] Franklin, M.S. et al. 2011. Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. *Psychonomic Bulletin & Review*. 18, 5 (2011), 992–997.

[17] Franklin, M.S. et al. 2013. Window to the wandering mind: pupillometry of spontaneous thought while reading. *The Quarterly Journal of Experimental Psychology*. 66, 12 (2013), 2289–2294.

[18] Holmes, G. et al. 1994. Weka: A machine learning workbench. *Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems* (1994), 357–361.

[19] Hutt, S. et al. 2016. The eyes have it: gaze-based detection of mind wandering during learning with an intelligent tutoring system. *Proceedings of the 9th International Conference on Educational Data Mining, International Educational Data Mining Society* (2016), 86–93.

[20] Jeni, L.A. et al. 2013. Facing Imbalanced Data–Recommendations for the Use of Performance Metrics. *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on* (2013), 245–251.

[21] Kononenko, I. 1994. Estimating attributes: analysis and extensions of RELIEF. *Machine Learning: ECML-94* (1994), 171–182.

[22] Kopp, K. et al. 2015. Influencing the occurrence of mind wandering while reading. *Consciousness and cognition*. 34, (2015), 52–62.

[23] Kopp, K. et al. 2015. Mind wandering during film comprehension: The role of prior knowledge and situational interest. *Psychonomic Bulletin & Review*. 23, 3 (2015), 842–848.

[24] Littlewort, G. et al. 2011. The computer expression recognition toolbox (CERT). *2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)* (2011), 298–305.

[25] Mills, C. et al. 2016. Automatic Gaze-Based Detection of Mind Wandering during Film Viewing. *Proceedings of the 9th International Conference on Educational Data Mining* (Raleigh, NC, USA, Jun. 2016).

[26] Pham, P. and Wang, J. 2015. AttentiveLearner: Improving Mobile MOOC Learning via Implicit Heart Rate Tracking. *Artificial Intelligence in Education*. C. Conati et al., eds. Springer International Publishing. 367–376.

[27] Randall, J.G. et al. 2014. Mind-Wandering, cognition, and performance: A theory-driven meta-analysis of attention regulation. *Psychological bulletin*. 140, 6 (2014), 1411.

[28] Reichle, E.D. et al. 2010. Eye movements during mindless reading. *Psychological Science*. 21, 9 (2010), 1300–1310.

[29] Risko, E.F. et al. 2013. Everyday attention: Mind wandering and computer use during lectures. *Computers & Education*. 68, (2013), 275–283.

[30] Smallwood, J. et al. 2004. Subjective experience and the attentional lapse: Task engagement and disengagement during sustained attention. *Consciousness and cognition*. 13, 4 (2004), 657–690.

[31] Wan, X. 2009. Co-training for Cross-lingual Sentiment Classification. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (Stroudsburg, PA, USA, 2009), 235–243.

[32] Webb, N. and Ferguson, M. 2010. Automatic Extraction of Cue Phrases for Cross-corpus Dialogue Act Classification. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (Stroudsburg, PA, USA, 2010), 1310–1317.

[33] Westlund, J.K. et al. 2015. Motion Tracker: Camera-Based monitoring of bodily movements using motion silhouettes. *PloS one*. 10, 6 (2015).

[34] Zacks, J.M. et al. 2010. The brain's cutting-room floor: Segmentation of narrative cinema. *Frontiers in human neuroscience*. 4, 168 (2010), 1–15.

[35] Zhang, Z. et al. 2011. Unsupervised learning in cross-corpus acoustic emotion recognition. *2011 IEEE Workshop on Automatic Speech Recognition Understanding* (Dec. 2011), 523–528.

[36] 2016. *Emotient module: Facial expression emotion analysis*.

Addressing Student Behavior and Affect with Empathy and Growth Mindset

Shamya Karumbaiah
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01003-9264
shamya@cs.umass.edu

Rafael Lizarralde
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01003-9264
rezecib@cs.umass.edu

Danielle Allessio
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01003-9264
allessio@umass.edu

Beverly Woolf
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01003-9264
bev@cs.umass.edu

Ivon Arroyo
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA 01609
iarroyo@wpi.edu

Naomi Wixon
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA 01609
mwixon@wpi.edu

ABSTRACT

We present results of a randomized controlled study that compared different types of affective messages delivered by pedagogical agents. We used animated characters that were empathic and emphasized the malleability of intelligence and the importance of effort. Results showed significant correlations between students who received more *empathic messages* and those who were *more confident, more patient*, exhibited *higher* levels of *interest*, and *valued* math knowledge more. Students who received more *growth mindset* messages, *tended to get more problems correct* on their first attempt but *valued* math knowledge *less* and had lower *posttest scores*. Students who received more *success/failure* messages tended to make *more mistakes*, to be *less learning-oriented*, and stated that they were *more confused*. We conclude that these affective messages are powerful media to influence students' perceptions of themselves as learners, as well as their perceptions of the domain being taught. We have reported significant results that support the use of empathy to improve student affect and attitudes in a math tutor.

Keywords

student affect, empathy messages, growth mindset, pedagogical agents, intelligent tutor, confidence

1. INTRODUCTION

Students experience many emotions while studying and taking tests [16]. Students' emotions (such as confidence, boredom, and anxiety) can influence achievement outcomes [10, 18] and predispositions (such as low self-concept and pessimism) can diminish academic success [5, 14].

Pekrun's Control-Value Theory of emotion has been experimentally validated by classroom experiments that used student self-reports (answers to 5-point scale survey questions). These experiments provide evidence that educational interventions can reduce students' anxiety [16, 19].

Dweck's Growth Mindset Theory suggests that students who believe that intelligence can be increased through effort and persistence tend to seek out academic challenges, compared to those who view their intelligence as immutable [8, 9]. Students who are praised for their effort (as opposed to performance) are more likely to view intelligence as being malleable, and their self-esteem remains stable regardless of how hard they have to work to succeed at a task.

Hattie and Timperley [13] studied which types of feedback and conditions enable learning to flourish and which cases stifle growth. According to their study feedback is intended to help a student get from where they are to where they need to be. Graesser et al., [12] reported that there are significant relationships between the content of feedback dialogue and the emotions experienced during learning. They found significant correlations between dialog and the affective states of confusion, eureka (delight) and frustration.

Pekrun et al., [17] tested a theoretical model positing that a student's anticipated achievement feedback in a classroom setting influences his/her achievement goals and emotions. For example, *self-referential feedback*, in which a student's competence is defined in terms of self-improvement, had a positive influence on a student's mastery goal adoption. On the other hand, *normative feedback*, in which student competence is defined relative to other students' mastery goals and performance goals, had a positive influence on *performance-approach* and *performance-avoidance* goal adoption. Furthermore, feedback condition and achievement goals predicted test-related emotions (i.e., enjoyment, hope, pride, relief, anger, anxiety, hopelessness, and shame).

Teachers have limited opportunities to recognize and respond to individual student's affect in typical classrooms.

Ideally, digital learning environments can manage the delicate balance between motivation and cognition, promoting both interest and deep learning. The overwhelming majority of work on affect-aware virtual tutors has focused on modeling affect, i.e., designing computational models capable of detecting how students feel while they interact with intelligent tutoring systems [2]. While modeling affect is a critical first step, very little research exists on systematically exploring the impact of interventions on students' performance, learning, and attitudes, i.e., how an environment might respond to students emotions (e.g., frustration, anxiety, and boredom) as they arise. D'Mello and Graesser carried out close research work on empathic characters in AutoTutor, a conversational tutor that uses 3D companions to conduct dialogs in natural language with students [6, 7, 11].

1.1 MathSpring

The testbed for this research is MathSpring, an intelligent tutor that personalizes mathematics problems, provides help using multimedia, and effectively teaches students to improve in standardized test scores [4]. Learning companions (Figure 1) in MathSpring suggest to students that their effort contributes to success, and that making mistakes only means more effort is needed. Companions use about 20 different messages focused on effort and growth mindset (Table 2).

To date, MathSpring learning companions have provided positive significant effects for the overall population of students and were more effective for lower achieving students and for female students in general [2]. However, characters seemed to have been harmful to some students (e.g., high-achieving males), who had higher affective baselines at pretest time and seem to have been distracted by the characters. These results suggest that affective characters should probably be different for students who are not presently frustrated or anxious (often high achieving males). One possibility is that the behavior of the characters be adaptive to the affective state of the student.

1.2 Recognize and Respond to Affect

Previously, we evaluated the hypothesis that **tailored affective messages delivered by digital animated characters may positively impact students emotions, attitude, and learning performance**. Specifically, we identified concrete prescriptive principles about how to respond to student emotion as it occurs during online learning [1, 3]. With models of student emotion, we explored mechanisms to address negative emotions. Our models predict confidence, interest, frustration, and excitement in real-time, based on data from hundreds of students. The gold standard was students' self-reported responses to questions, such as "How confident do you feel right now?"

We found that **growth mindset messages** based on Dweck's theory [9] provide an apparent **boost in student math learning** [3], resulted in **less performance-oriented goals** (e.g., beating classmates, instead of a self-referenced focus), and **less boredom** reported on the posttest. Typically online educational systems only report correctness: "Your answer is correct/incorrect." We discovered that such **success/failure** messages are correlated to higher reported **anxiety** and **boredom**, and appear to increase **performance-oriented goals**[3]. Other results indicate that empathic characters can help decrease students' anxiety and boredom. Our results showed that: a) student anxiety and boredom can be reduced using simple 2D characters, as did D'Mello et al., (2007); b) these benefits are due primarily to empathy, and secondarily to growth mindset messages; and c) indicating only success or failure is actively **harmful** to students, in comparison to emphasizing the learning process and the importance of effort.

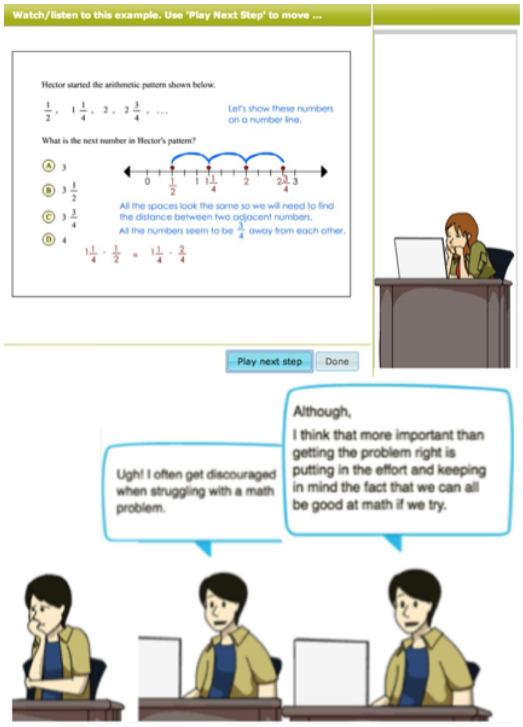


Figure 1: Learning companions respond to student actions with gestures and messages shown both as text and audio. *Above:* Companion shows high interest while the student views an example problem with solution steps shown. *Below:* Companion provides a growth mindset message, encouraging the student to put in effort to become good at math.

1.3 Research Goals

The research questions in this paper focus on identifying messages that support students' motivation to persist working on a task. Which messages (see Table 2) should a tutoring system send to students to encourage them to persist? How should agents respond to negative emotions? Should students be praised when they do well? Are the benefits to student learning and emotion due to empathic or motivational aspects of the companion? What are the results on learning and emotion of using an empathic or less empathic companion in comparison to a companion that indicates only success or failure?

Table 1: Outcomes variables measured in the experiment. The questions on the pre- and posttest were answered in a 5-point scale, going from "not at all" to "very much".

Interest - Students' interest in math. "Are you interested when solving math problems?"
Excitement - How exciting students find math. "Do you feel that solving math is exciting?"
Confusion - How confused students feel while solving math problems. "Do you feel confident that you will eventually be able to understand the Mathematics material?"
Frustration - How frustrating students find math. Average of "Do you get frustrated when solving math problems?" and "Does solving math problems make your feel frustrated?"
Learning Orientation - How much students focus on learning as opposed to performance. Average of "When you are doing math exercises, is your goal to learn as much as you can?" and "Do you prefer learning about things that make you curious even if that means you have to work harder?"
Performance Approach Goals - "Do you want to show that you are better at math than your classmates?"
Math Value - How important do students think math is. "Compared to most other activities, how important is it or you to be good at math?"
Math Liking - Measure of how much students like math. "Do you like your math class?"
Math Test Performance - Student's score on math questions that are representative of the content covered in MathSpring.

2. METHOD

We conducted a randomized controlled study to evaluate three different types of affective messages delivered by pedagogical agents (Table 2). The study took place in an urban school district in Southern California with sixty-four 6th grade students in three math classes for four class sessions, during December 2016. On part of the first and last day, students completed a pretest and posttest including questions related to various affective states, and questions about mathematics. Outcome variables measured from these questions are provided in Table 1.

Three conditions of learning companion messages were randomly assigned to students and delivered in both audio and written form in order to increase the likelihood of exposure: 1) **Empathy Condition** for 24 students, 2) **Growth Mindset Condition** for 20 students and 3) **Success/Failure Condition** for 20 students; see Table 2 for examples of the different types of messages. For all conditions, students were asked to self-report their frustration or confidence in a five-point scale every five minutes or every eight problems, which ever came first, but only after a problem was completed. The prompts were shown on a separate screen and invited students to report on their frustration or confidence.

The **Empathy** condition was set to visually reflect positive emotion with a certain probability for each math problem if the last student emotion report had a positive valence. When the most recent emotion report had a negative valence, and with a certain probability, the character first visually reflected the negative emotion; then it reported an empathy message such as "Sometimes these problems make me feel [frustrated]", and finally a connector such as "on the other hand", connected with a growth mindset message such as "I know that putting effort into problem solving and learning from hints will make our intelligence grow." Note that only students experiencing negative emotions were exposed to growth mindset messages, as opposed to the following condition.

The **Growth Mindset** condition emphasized messages that accentuate the importance of effort and perseverance in achieving success. The growth mindset condition was set to provide one of many growth mindset messages after a second incorrect attempt was made (the first incorrect attempt caused the hint button to flash), regardless of students' emotions. This condition also provided occasional growth mindset messages at the beginning of a new problem.

The **Success/Failure** condition provided both traditional success/failure messages and some more basic meta-cognitive support for when students made mistakes (e.g., acknowledging that their answer was not correct while encouraging them to use a hint). The success/failure condition provided students with a response if they answered a problem correctly and also after they made a second mistake.

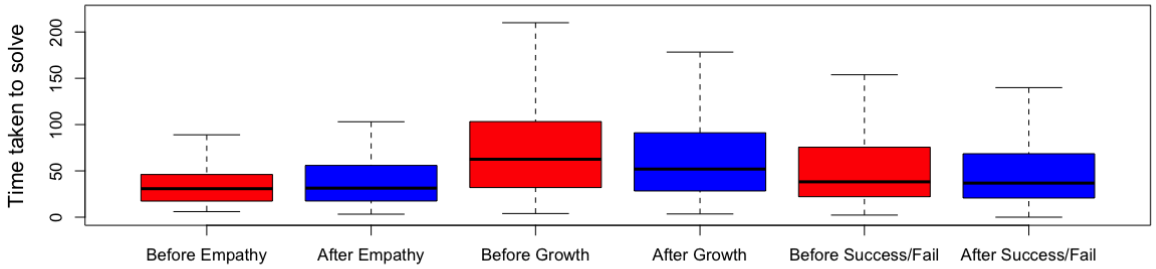
3. RESULTS

Out of the 64, three students' data were discarded due to minimal interaction with MathSpring. Across the $N = 61$ students, 21066 event log rows were recorded for three classes over four separate days, from which several behavioral features were derived and used throughout the analysis; our data and processing scripts can be found on GitHub [15]. All the students completed a pretest and posttest. Students in empathy, growth mindset and success/failure conditions received a total of 978, 763, and 882 messages respectively. Means, standard deviations and percentage shares for each type of message are given in Table 3. It is important to note that students received messages from all categories but their condition emphasized the corresponding message type. For example, a student in growth mindset condition received significantly more growth mindset messages than a student in empathy condition. This distribution of messages means that different students saw different amounts of each type of message, which allows us to perform partial correlations with respect to the counts of each message type, separating their effects.

Table 2: Examples of messages spoken by characters.

Condition	Message
Empathy	"Don't you sometimes get frustrated trying to solve math problems? I do. But guess what. Keep in mind that when you are struggling with are new idea or skill you are learning something and becoming smarter."
Growth Mindset	"Hey, congratulations! Your effort paid off, you got it right!" "Did you know that when we practice to learn new math skills our brain grows and gets stronger?" "Let's click on help, and I am sure we will learn something."
Success/ Failure	"Very good, we got another one right!" "Hmm. Wrong. Shall we work it out on paper?"

Figure 2: Time spent on a problem immediately before and after receiving the different categories of messages.



3.1 Partial Correlations

First, we attempted to replicate the results of our previous exploratory work [3]. For the three message types, partial correlations of the total number of each messages were measured for the nine posttest measures, controlling for the corresponding pretest measure, time spent in the tutor, and message frequency (total messages heard / time spent).

Table 4 shows the result of this analysis. We observe that with exposure to more **empathic** messages, students exhibited **higher levels of interest** and **valued math knowledge more** (rows 1 and 7). Increased interest can be viewed as analogous to the high negative correlation with boredom reported in our earlier work. With **growth mindset** messages, students **valued math knowledge less** and had **lower post test performance scores** (rows 7 and 9). With **success/failure** messages, students were **less learning-oriented** and claimed to be **more confused** (rows 6 and 3).

To further understand the dynamics, we derived some in-tutor variables and performed partial correlations shown in Table 5. The data for this analysis was derived as per student metrics based on their interaction with MathSpring. We observed that students tend to answer significantly more questions when in the **success/failure** condition and end up making more mistakes as well (rows 4 and 5). It is important to note that they also **avoid asking for hints** (row 6). It seems like these students tend to rush through the problems while being more careless. They also make **more mistakes** when they receive more growth mindset messages (row 5). This leads to simpler questions which they tend to get right in the first attempt (row 1). It appears that the students in **empathy** condition continue to **invest more time** on solving problems than rushing through the problem set. The number of problems seen by these students is significantly less (row 4).

As we see in Figure 2, students tend to spend less time on problems immediately after they receive growth mindset or success/failure messages. In contrast, the time spent on a problem increases slightly after receiving empathic messages. Students who received more empathic and growth mindset messages tend to answer fewer questions than do students who received mostly success/failure message (Figure 3). Combined with the last plot, it looks like the students in the empathy condition continue to invest more time on solving problems than rushing through the problem set.

Figure 3: Problems seen per minute across different pedagogies

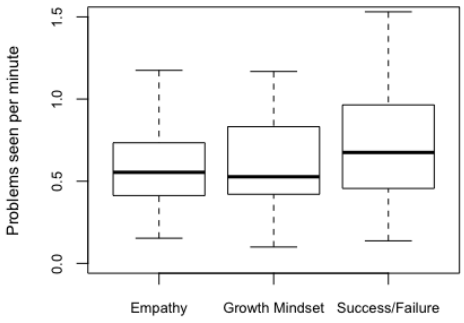


Table 3: The distribution of messages seen by students in each pedagogical conditions.

Condition	N	Empathy Messages			Growth Mindset Messages			Success/Failure Messages		
		mean	std	%	mean	std	%	mean	std	%
Empathy	21	7.48	7.0	16%	9.95	7.2	21%	29.1	22	62%
Growth Mindset	20	0.2	0.5	0.5%	10	5	26%	27.9	19.2	73%
Success/ Failure	20	1.2	1.7	2.7%	4.6	4.8	10%	38.3	26.6	86%

Table 4: Partial correlations between different types of messages seen and posttest variables (Table 1), accounting for the corresponding pretest value, time spent in tutor and message frequency.

	Variable	Empathy Messages		Growth Mindset Messages		Success/Failure Messages	
		corr	p	corr	p	corr	p
(1)	**Interest**	**0.28***	0.03	0.19	0.15	-0.20	0.14
(2)	**Excitement**	0.00	1.00	-0.07	0.60	-0.08	0.54
(3)	**Confusion**	-0.05	0.74	-0.05	0.74	**0.32***	0.02
(4)	**Frustration**	0.10	0.43	-0.08	0.54	-0.18	0.18
(5)	**Performance Approach**	-0.19	0.14	-0.05	0.70	0.20	0.12
(6)	**Learning Orientation**	0.02	0.85	0.02	0.88	**-0.24**$^{+}$	0.06
(7)	**Math Value**	**0.25***	0.05	**-0.22**$^{+}$	0.09	-0.10	0.45
(8)	**Math Liking**	0.01	0.96	0.01	0.96	0.05	0.72
(9)	**Performance**	-0.01	0.93	**-0.23**$^{+}$	0.07	-0.13	0.33

$^{+}$ $p \leq 0.10$, * $p \leq 0.05$

Table 5: Partial correlations between different types of messages seen and within-tutor variables, accounting for time spent in the tutor and message frequency.

	Variable	Empathy Messages		Growth Mindset Messages		Success/Failure Messages	
		corr	p	corr	p	corr	p
(1)	**% Problems Solved on First Attempt**	0.06	0.62	**0.34****	0.007	-0.01	0.94
(2)	**Avg Problem Difficulty**	0.07	0.61	-0.05	0.69	0.19	0.14
(3)	**Learning Gain**	-0.10	0.50	-0.07	0.63	-0.14	0.34
(4)	**Problems Seen**	**-0.23**$^{+}$	0.07	-0.04	0.78	**0.77****	4E-13
(5)	**Mistakes Made**	-0.01	0.91	**0.59****	6E-7	**0.30***	0.02
(6)	**Hints Per Problem**	0.10	0.43	0.16	0.22	**-0.22**$^{+}$	0.10

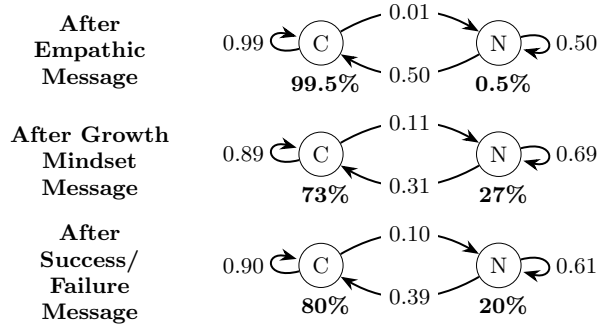
$^{+}$ $p \leq 0.10$, * $p \leq 0.05$, ** $p \leq 0.01$

3.2 Markov Chain Analysis

As students solve problems in the tutoring system, the learning companion comments on their attempts; the effect of these messages on student affect is sequential, but the partial correlations do not capture this. To analyze this effect, we built Markov Chain models using in-tutor student self-reports of confidence and frustration. Each model describes transitions in affective states, from one self-report to the next, where students received a particular type of character messages (empathy, growth mindset, and success/failure) between self-reports. To reduce the state space, the 5-point scale used in the self-reports was simplified to two values - confident (≥ 3), not confident (< 3); similarly for frustration.

The goal of the Markov models was not to predict emotional changes, but rather to examine whether different messages had significant effects on affect. Markov models can show the probability of transitioning between affective states, but also have a stationary distribution, which represents the amount of students that would be in each state after undergoing many transitions. For example, a group of students were to use the system for many hours and receive only empathic messages, our model suggests that 99.5% of them would be confident about learning math (Figure 4).

Figure 4: State transitions between the Confident (C) and Not Confident (N) affective states. The stationary distribution is shown below each state. Only the empathy model was significant in the likelihood ratio test ($p \leq 0.05$)



We used a likelihood ratio test to analyze the significance of these models: the probability of the null model (ignoring message type) generating the data divided by the probability of the alternate model (for a particular message type) generating the data gives a p-value. Figure 4 shows the state transitions for **confidence** in the null model and the model for confidence after receiving **empathic** messages, which was significant with $p = 0.0149$ (the other models were not significant). We also examined the stationary distributions for each model (Table 6).

Table 6: Stationary distributions in the Markov models of confidence and frustration.

Message Type	Confidence		Frustration	
	Conf	Not	Frust	Not
Empathy	**99.5%***	**0.05%***	35%	65%
Growth Mindset	74%	26%	30%	70%
Success/ Failure	80%	20%	25%	75%

$^*p \leq 0.05$

4. DISCUSSION

Some of our results support the hypothesis that affective messages delivered by characters can positively impact students' emotions and affective predispositions for math problem solving. This is particularly evident for empathy, as the more empathic messages a student saw the higher their interest in mathematics problem solving, as well as their beliefs that mathematics is valuable to learn (Table 4). An analysis of student behavior suggests that students who saw a high frequency of empathic messages also tended to be more patient and cautious with problem solving, suggesting that empathic messages may encourage students to persist through adversity. Exposure to empathic messages was significantly correlated to investing time in each math problem activity, leading also to fewer problems seen per session. A positive trend is exhibited between high frequency of empathic messages and hints requested, even if not significant (Table 5). Empirical temporal models generated from students' changes in self-reports of affect, within the tutor, revealed that students receiving empathic messages have a higher likelihood to become more confident and to remain confident.

The response to growth mindset messages delivered by characters yielded mixed results. As students saw more of these kinds of messages they also succeeded more often at solving problems correctly (on the first attempt); interestingly, at the same time, they also made more mistakes. This is also desirable, as growth mindset messages emphasize that making mistakes is okay and can even help learning, legitimizing a high frequency of errors. It is possible that students were using those mistakes and hints to learn and succeed later on; a (not significant) positive trend suggests that students receiving more of these kinds of messages also asked for more hints per problem. In contrast, marginally significant effects suggest that a high frequency of growth mindset messages might be detrimental to students' perception of math value, and that their posttest performance is worse when they receive more of this kind of messages. It is hard to conclude the meaning of these marginally significant effects, especially because a previous study suggested that these messages were beneficial in general [3]. Note that empathic messages used 'growth mindset' messages also, in order to resolve the negative emotion (see Table 2). One possible explanation is that the empathic condition was so positive because it was also very selective at showing growth mindset messages for only those who experienced negative emotions. It is likely that high achieving students, or those who "felt OK", rejected growth mindset messages that they might have perceived to be unnecessary.

An important comment is that we did not expect that success/failure messages could be so harmful to students. Regardless of whether messages indicated success or failure, as students received more of these messages they also exhibited lower levels of mastery/learning orientation at posttest time. They also reported higher levels of confusion at posttest time (note that the confusion can be positive for learning within the learning experience, but not after the learning experience has concluded). Regarding behavior within the tutor, the more students were exposed to success/failure messages, the more they appeared to rush through problems, make mistakes, and request fewer hints per problem.

To summarize, empathy messages were associated with variables consistent with methodical work and an increased interest/value of mathematics. However, both growth mindset and success/failure messages appeared to be associated with a greater number of mistakes. Finally, success/failure messages themselves were associated with a whole host of concerning behaviors such as confusion with the material following posttest, reduced learning orientation, hurried work, and a reduced likelihood of requesting hints. This is consistent with Dweck's findings that growth mindset messages are superior to success/failure messages [8, 9]. Whether empathic messages in fact result in improved student performance pre to posttest will likely require larger samples than this small study (N = 61). However, students in non-empathic conditions have demonstrated significantly more mistakes in their work.

5. CONCLUSIONS

This research emphasizes the importance of understanding an intervention's effect on a student's affective state, which in turn is connected to engagement, performance, and learning. Although many researchers have focused on modeling affect, very little research effort has been put into systematically measuring the impact of the intervention on the student behavior in an adaptive learning environment. Empathic messages that respond to students' recent emotions have resulted in superior results both in improving the student interaction with the system and in the overall learning experience. Growth Mindset follows next with some positive impact on in-tutor performance but its overall effect in the short-term is questionable. Success/Failure messages are generally harmful to students: reducing learning orientation, increasing confusion, and making students more careless during the learning experience.

We conclude that affective messages delivered by characters in online tutoring environments are a very important medium for building student-tutor rapport in a virtual environment, powerful signals that influence perceptions of students themselves as learners, as well as perceptions of the domain being taught. We have reported significant results that support the use of empathy to improve student affect and attitudes in a math tutor. The long-term effect of these messages needs to be studied when the novelty of this intervention wears off. In the future, we hope to study the impact of the frequency and content of these messages. To evaluate the generalizability of these results, student populations across different demographics needs to be studied as well as the applicability of the messages to domains beyond mathematics.

6. ACKNOWLEDGMENTS

This research is supported by the National Science Foundation (NSF) 1324385 IIS/Cyberlearning DIP: Collaborative Research: Impact of Adaptive Interventions on Student Affect, Performance, and Learning. Any opinions, findings, and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

7. ADDITIONAL AUTHORS

Additional authors: Winslow Burleson (New York University, 70 Washington Square South New York, New York, 10012; email: wb50@nyu.edu).

8. REFERENCES

[1] I. Arroyo, W. Burleson, M. Tai, K. Muldner, and B. P. Woolf. Gender differences in the use and benefit of advanced learning technologies for mathematics. *Journal of Educational Psychology*, 105(4):957, 2013.

[2] I. Arroyo, D. G. Cooper, W. Burleson, B. P. Woolf, K. Muldner, and R. Christopherson. Emotion sensors go to school. In *AIED*, volume 200, pages 17–24, 2009.

[3] I. Arroyo, S. Schultz, N. Wixon, K. Muldner, W. Burleson, and B. P. Woolf. Addressing affective states with empathy and growth mindset. *6th International Workshop on Personalization Approaches in Learning Environments*, 2016.

[4] I. Arroyo, B. P. Woolf, W. Burelson, K. Muldner, D. Rai, and M. Tai. A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect. *International Journal of Artificial Intelligence in Education*, 24(4):387–426, 2014.

[5] L. Corno and R. E. Snow. Adapting teaching to individual differences among learners. *Handbook of research on teaching*, 3(605-629), 1986.

[6] S. D'Mello and A. Graesser. Automatic detection of learner's affect from gross body language. *Applied Artificial Intelligence*, 23(2):123–150, 2009.

[7] S. D'Mello and A. Graesser. Autotutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(4):23, 2012.

[8] C. S. Dweck. *Self-theories: Their role in motivation, personality, and development*. Psychology Press, 2000.

[9] C. S. Dweck. Beliefs that make smart people dumb. *Why smart people can be so stupid*, 24:41, 2002.

[10] D. Goleman. Emotional intelligence. why it can matter more than fq. *Learning*, 24(6):49–50, 1996.

[11] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618, 2005.

[12] A. C. Graesser, S. K. D'Mello, S. D. Craig, A. Witherspoon, J. Sullins, B. McDaniel, and B. Gholson. The relationship between affective states and dialog patterns during interactions with autotutor. *Journal of Interactive Learning Research*, 19(2):293, 2008.

[13] J. Hattie and H. Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.

[14] A. N. Kluger and A. DeNisi. Feedback interventions: Toward the understanding of a double-edged sword. *Current directions in psychological science*, 7(3):67–72, 1998.

[15] R. Lizarralde and S. Karumbaiah. A collection of scripts for processing mathspring data. `https://github.com/rezecib/MathspringDataProcessing`, 2017.

[16] R. Pekrun. Emotions and learning. *International Academy of Education. Australia: International Bureau of Education*, 2014.

[17] R. Pekrun, A. Cusack, K. Murayama, A. J. Elliot, and K. Thomas. The power of anticipated feedback: Effects on students' achievement goals and achievement emotions. *Learning and Instruction*, 29:115–124, 2014.

[18] R. Pekrun, T. Goetz, L. M. Daniels, R. H. Stupnisky, and R. P. Perry. Boredom in achievement settings: Exploring control-value antecedents and performance outcomes of a neglected emotion. *Journal of Educational Psychology*, 102(3):531, 2010.

[19] R. Pekrun, E. Vogl, K. R. Muis, and G. M. Sinatra. Measuring emotions during epistemic activities: the epistemically-related emotion scales. *Cognition and Emotion*, pages 1–9, 2016.

Epistemic Network Analysis and Topic Modeling for Chat Data from Collaborative Learning Environment

Zhiqiang Cai
The University of Memphis
365 Innovation Drive, Suite 410
Memphis, TN, USA
zcai@memphis.edu

Brendan Eagan
University of Wisconsin-Madison
1025 West Johnson Street
Madison, WI, USA
eaganb@gmail.com

Nia M. Dowell
The University of Memphis
365 Innovation Drive, Suite 410
Memphis, TN, USA
niadowell@gmail.com

James W. Pennebaker
University of Texas-Austin
116 Inner Campus Dr Stop G6000
Austin, TX, USA
pennebaker@utexas.edu

David W. Shaffer
University of Wisconsin-Madison
1025 West Johnson Street
Madison, WI, USA
dws@education.wisc.edu

Arthur C. Graesser
The University of Memphis
365 Innovation Drive, Suite 403
Memphis, TN, USA
art.graesser@gmail.com

ABSTRACT

This study investigates a possible way to analyze chat data from collaborative learning environments using epistemic network analysis and topic modeling. A 300-topic general topic model built from TASA (Touchstone Applied Science Associates) corpus was used in this study. 300 topic scores for each of the 15,670 utterances in our chat data were computed. Seven relevant topics were selected based on the total document scores. While the aggregated topic scores had some power in predicting students' learning, using epistemic network analysis enables assessing the data from a different angle. The results showed that the topic score based epistemic networks between low gain students and high gain students were significantly different ($t = 2.00$). Overall, the results suggest these two analytical approaches provide complementary information and afford new insights into the processes related to successful collaborative interactions.

Keywords

chat; collaborative learning; topic modeling; epistemic network analysis

1. INTRODUCTION

Collaborative learning is a special form of learning and interaction that affords opportunities for groups of students to combine cognitive resources and synchronously or asynchronously participate in tasks to accomplish shared learning goals [15; 20]. Collaborative learning groups can range from a pair of learners (called a dyad), to small groups (3-5 learners), to classroom learning (25-35 learners), and more recently large-scale online learning environments with hundreds or even thousands of students [5; 22]. The collaborative process provides learners with a more efficient learning experience and improves learners' collaborative learning skills, which are critical competencies for students [14]. Members in a team are different in many ways. They have their own experience, knowledge, skills, and approaches to learning. A student in a collaborative learning environment can take other students' views and ideas about the information provided in the learning material. The ideas coming out of the team can then be integrated as a deeper understanding of the material, or a better solution to a problem.

Traditional collaborative learning occurred in the form of face to face group discussion or problem solving. As the internet and learning technologies develop, online collaborative learning environments come out and are playing more and more important roles. For example, MOOCs (Massive Open Online Courses) have drawn massive number of learners. Learners in MOOCs are connected by the internet and can easily interact with each other using various types of tools, such as forums, blogs and social networks [23]. These digitized environments make it possible to track the learning processes in collaborative learning environments in greater detail.

Communication is one of the main factors that differentiates collaborative learning from individual learning [4; 6; 9]. As such, chats from collaborative learning environments provide rich data that contains information about the dynamics in a learning process. Understanding massive chat data from collaborative learning environments is interesting and challenging. Many tools have been invented and used in chat data analysis, such as LIWC (linguistic inquiry and word count) [12], Coh-Metrix [10], and topic modeling, just to name a few. Epistemic network analysis (ENA) has been playing a unique role in analyzing chat data from epistemic games [18]. ENA is rooted in a specific theory of learning: the epistemic frame theory, in which the collection of skill, knowledge, identity, value and epistemology (SKIVE) forms an epistemic frame. A critical theoretical assumption of ENA is that the connections between the elements of epistemic frames are critical for learning, not their presence in isolation. The online ENA toolkit allows users to analyze chat data by comparing the connections within the epistemic networks derived from chats. ENA visualization displays the clustering of learners and groups and the network connections of individual learners and groups. ENA requires coded data which has traditionally relied on hand coded data sets or classifiers that rely on regular expression mapping. Combining topic modeling with ENA will provide a new mode of preparing data sets for analysis using ENA.

In this study, we used a combination of topic modeling and ENA to analyze chat data to see if we could detect differences between the connections made by students with high learning gains versus students with low learning gains. Incorporating topic modeling

with ENA will make the analytic tool more fully automated and of greater use to the research community.

2. RELATED WORK

Chats have two obvious features. First, they appear in the form of text. Therefore, any text analysis tool may have a role in chat analysis. Second, chats come from individuals' interaction, which reflects social dynamics between participants. Therefore, a combination of text analysis and social network analysis should be helpful in understanding underlying chat dynamics. For instance, Tuulos et al. [21] combined topic modeling with social network analysis in chat data analysis. They found that topic modeling can help identify the receiver of chats (the person who a chat is given to).

In a similar effort, Scholand et al. [16] combined LIWC and social network analysis to form a method called "social language network analysis" (SLNA). The social networks were formed by counting the number of times chat occurred between any two participants. Based on the counts, participants were clustered into a tree structure, representing the level of subgroups the participants belong to. LIWC was then used to get the text features of chats. It was found that, some LIWC features were significantly different between in group conversations and out of group conversations.

Researchers have also recently explored the advantages of combining SNA (social network analysis) with deeper level computational linguistic tools, like Coh-Metrix. Coh-Metrix computes over 100 text features. The five most important Coh-Metrix features are: narrativity, syntax simplicity, word concreteness, referential cohesion and deep cohesion. Dowell and colleagues [8] explored the extent to which characteristics of discourse diagnostically reveals learners' performance and social position in MOOCs. They found that learners who performed significantly better engaged in more expository style discourse, with surface and deep level cohesive integration, abstract language, and simple syntactic structures. However, linguistic profiles of the centrally positioned learners differed from the high performers. Learners with a more significant and central position in their social network engaged using a more narrative style discourse with less overlap between words and ideas, simpler syntactic structures and abstract words. An increasing methodological contribution of this work highlights how automated linguistic analysis of student interactions can complement social network analysis (SNA) techniques by adding rich contextual information to the structural patterns of learner interactions.

In another study, Dowell et al. [7] showed that students' linguistic characteristics, namely higher degrees of narrativity and deep cohesion, are predictive of their learning. That is, students engaged in deep cohesive interactions performed better.

In the present research, we explore collaborative interaction chat data using the combination of topic modeling and epistemic network analysis. While previous studies focused on the relationship between language features and social network connections, our study focuses on prediction learning performance by semantic network connections students make in chats.

3. METHODS

Participants. Participants were enrolled in an introductory-level psychology course taught in the Fall semester of 2011 at a large university in the USA. While 854 students participated in this course, some minor data loss occurred after removing outliers and those who failed to complete the outcome measures. The final sample consisted of 844 students. Females made up 64.3% of this final sample. Within the population, 50.5% of the sample identified as Caucasian, 22.2% as Hispanic/Latino, 15.4% as Asian American, 4.4% as African American, and less than 1% identified as either Native American or Pacific Islander.

Course Details and Procedure. Students were told that they would be participating in an assignment that involved a collaborative discussion on personality disorders and taking quizzes. Students were told that their assignment was to log into an online educational platform specific to the University at a specified time, where they would take quizzes and interact via web chat with one to four random group members. Students were also instructed that, prior to logging onto the educational platform, they would have to read material on personality disorders. After logging into the system, students took a 10 item, multiple choice pretest quiz. This quiz asked students to apply their knowledge of personality disorders to various scenarios and to draw conclusions based on the nature of the disorders. The following is an example of the types of quiz questions students were exposed to:

- *Jacob was diagnosed with narcissistic personality disorder. Why might Dr. Simon think this was the wrong diagnosis?*
- *Dr. Level has measured and described his 10 mice of varying ages in terms of their length (cm) and weight (g). How might he describe them on these characteristics using a dimensional approach?*
- *Danielle checks her facebook page every hour. Does Danielle have narcissistic personality disorder?*

After completing the quiz, they were randomly assigned to other students who were waiting to engage in the chatroom portion of the task. When there were at least 2 students and no more than 5 students (M = 4.59), individuals were directed to an instant messaging platform that was built into the educational platform. The group chat began as soon as someone typed the first message and lasted for 20 minutes. The chat window closed automatically after 20 minutes, at which time students took a second 10 multiple-choice question quiz. Each student contributed 154.0 words on average (SD = 104.9) in 19.5 sentences (SD = 12.5). As a group, discussions were about 714.8 words long (SD = 235.7) and 90.6 sentences long (SD = 33.5).

An excerpt of a collaborative interaction chat in a chat room is shown below in Table 1. (student names have been changed):

Table 1. An excerpt of a collaborative interaction chat

Student	*Chat Text*
Art	ok cool, everyone's here. sooo first question
Art	ok so the certain characteristics to be considered to have a personality disorder?
Shaffer	Alright sooo first question: Based on these criteria describe several reasons why a psychologist might not label someone with grandiose thoughts as having narcissistic personality disorder?
Shaffer	hahaha never mind
Shaffer	that was the second question.
Art	lol its all good
Shaffer	okay so certain characteristics: doesn't it have to be like a stable thing?
Carl	i think the main thing about having a disorder is that its disruptive socially and/or makes the person a danger to himself or others

Vasile	yes, stable over time
Shaffer	yeah, and it also mentioned it can't be because of drugs
Art	also they have to have like unrealistic fantasies
Nia	yeah and not normal in their culture
Carl	no drugs or physical injury
Vasile	begins in early adulthood or adolescence
Shaffer	i think that covers them? haha
Art	ok, so arrogance doesn't just define it, they have to have most of these characteristics
Art	yeah i think we got them
Shaffer	is it most or is it like 6?

From the above excerpt, we can see several obvious things. First, the lengths of the utterances varied from one single word to multiple sentences. This needs to be considered in text analysis because some methods work only for longer texts. For example, Coh-Metrix usually works well for texts with more than 200 words. Topic modeling also needs enough length to reliably infer topic scores. Second, the number of utterances each participant gave were different. From how much and what a member said, we can see each member played a different role in that chat. Third, the ordered sequence of the utterances forms a time series. Understanding and visualizing the underlying discourse dynamics are important for meaning making with this type of data.

The data set contained 15,670 utterances, pretest scores (the first quiz) and post test scores (the second quiz) for 844 students, grouped in 182 chat rooms. Each chat room had 2 to 5 students, 4.73 by average. The average speech turns each student gave was 18.2 and the average speech turns in each room was 86.1.

The average pretest score was 36.01% correct and the average post-test scores 45.73% correct. Paired sample test shows that the post-test is significantly higher ($t = 14.13, N = 844$). We computed the learning gain of each student, using the formula

$$gain = \frac{posttest\ score - pretest\ score}{1 - pretest\ score}.$$

For all students ($N = 844$), the average learning gain is 0.11, 59.5% had positive learning gains above 0.1. 16.5% had the same scores and 23% had negative learning gains. Not surprisingly, students who had lower pretest scores had higher learning gains because they had greater potential to learn. Figure 1 shows the average learning gain as function of pretest score.

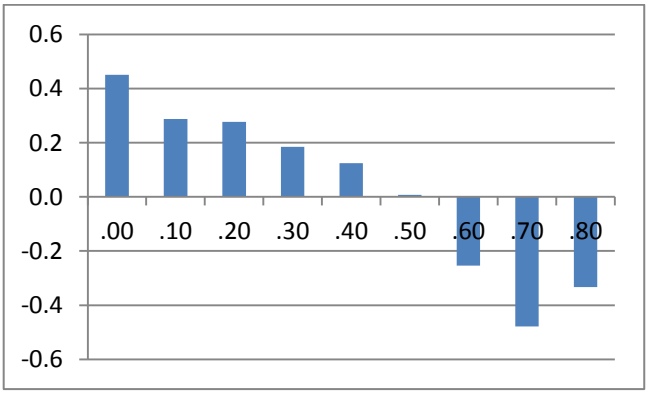


Figure 1. Average learning gain as a function of pretest score.

For students with pretest scores less than 50% correct (N=624), the average learning gain is 0.88, 69.7% had positive learning gains, 15.7% had the same scores and 14.6% had negative learning gains.

This data set has been analyzed in multiple studies. Cade et al. [3] analyzed the cohesion of the chats and found that deep cohesion of the chats predicts the students feeling of power and connectedness to the group. Dowell et al. [7] found that some Coh-Metrix measures predicts learning. Coh-Metrix measures describe common textual features that are not content specific. For example, cohesion is about how text segments are semantically linked to each other, which has nothing to do with what the text content is about. In this study, we use topic modeling to provide content dependent features and use epistemic network analysis to explore how the topics were associated in the chats.

4. TOPIC MODELING

Topic modeling has been widely used in text analysis to find what topics are in a text and what proportion/amount of each topic is contained. Latent Dirichlet Allocation (LDA) [2; 24] is one of the most popular methods for topic modeling. LDA uses a generative process to find topic representations. LDA starts from a large document set $D = \{d_1, d_2, \cdots, d_m\}$. A word list $W = \{w_1, w_2, \cdots, w_n\}$ is then extracted from the document set. LDA assumes that the document set contains a certain number of topics, say, *K* topics. Each document has a probability distribution over the *K* topics and each topic has a probability distribution over the given list of words. When a document was composed, each word that occurred in a document was assumed to be drawn based on the document-topic probability and the topic-word probability. For a given corpus (document set) and a given number of topics *K*, LDA can compute the topic assignment of each word in each document.

For a given topic, the word probability distribution can be easily computed from the number of times each word was assigned to the given topic. The beauty of topic modeling is that the "top words" (words with highest probabilities in a topic) usually give a meaningful interpretation of a topic. The distributions are the underlying representation of the topics. The top words are usually used to show what topics are contained in the corpus.

By counting the number of words assigned to each topic, a topic proportion score can be computed for each document on each topic. The topic proportion scores then become a document feature that can be used in further analysis. However, the proportion scores are based on the statistical topic assignment of words. When documents are very short, such as most utterances in our chat data, the topic proportion scores won't be reliable. Cai et al. [4] argued that alternative ways to compute document topic scores are possible.

4.1 TASA Topic Model

Although our chat data set contained 15,670 utterances, the utterances were short and the corpus is not large enough to build a reliable topic model. To get a reliable model, we used a well known corpus provided by TASA (Touchstone Applied Science Associates). This corpus contained documents on seven known categories, including business, health, home economics, industrial arts, language arts, science and social studies. Our content topic, personality disorders, is obviously in the health category. Of course, not all topics in TASA are relevant to our study. Therefore, after building up the model, we need to select relevant topics. We will cover that in the next sub-section.

There are a total of 37,651 documents in TASA corpus, each of which is about 250 words long. Before we ran LDA, we filtered out very high frequency words and very low frequency words. High frequency words, such as "the", "of", "in", etc., won't contain much topic information. Rare words won't contribute to meaningful statistics. 28,483 words (it might be better to say "terms") were left after filtering. A model with 300 topics was constructed by LDA.

4.2 Topic score computation and topic selection

From the TASA topic model, we computed the word-topic probabilities based on the number of times a word was assigned to each of the 300 topics. Thus, each word is represented by a 300 dimensional probability distribution vector. For each chat in our chat corpus, we simply summed up the word probability vectors for the words appeared in each chat. That gave us 300 topic scores for each chat. Recall that, the chats were associated with a reading material and two quizzes. While the students were free to talk about anything, the content of the reading material and the quizzes set up the main chat topics, that is, personality disorders.

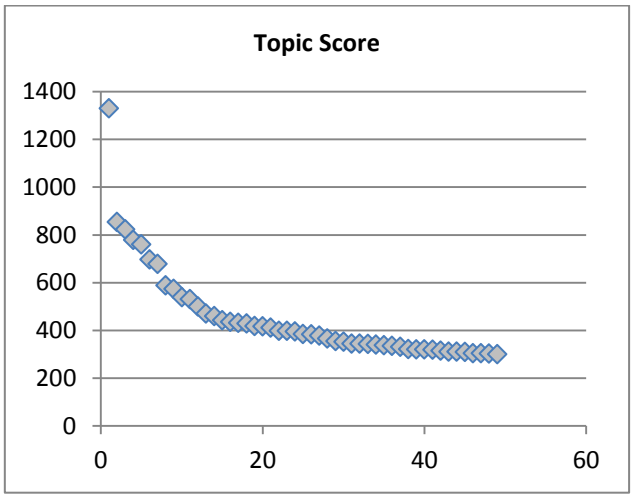


Figure 2. Sorted topic scores for topic selection.

The first thing we needed to do then was to investigate whether or not the "hot" topics from the computation made sense. To find that out, we computed the sum of all topic scores over all chats. The topics were sorted according the total topic score. The hottest topic had a total score higher than 1300, much higher than the second highest (less than 900). By examining the top words, this topic is about "illness", which is highly relevant to personality disorders. Six hot topics scored in the range from 600 to 900. They are about "outdoors", "biology", "people/social", "education" and "healthcare". The top words are listed below.

- **Illness:** health, disease, patient, body, diseases, medical, stress, mental, physical, heart, doctor, problems, cause, person, patients, exercise, illness, problem, nurse, healthy
- **Outdoors:** dog, energy, plants, earth, car, light, food, heat, words, animals, music, rock, language, children, air, uncle, city, sun, women, plant
- **Biology:** cells, cell, genes, chromosomes, traits, color, organisms, sex, egg, species, gene, body, male, female, parents, nucleus, eggs, sperm, organism, sexual
- **Psychology:** behavior, learning, theory, environment, feelings, sexual, physical, social, sex, human, research, person, animal, mental, response, positive, stress, personality, subject, reaction
- **People/Social:** joe, pete, mr, charlie, dad, frank, billy, tony, jerry, 'll, mom, 'd, going, 're, got, boys, looked, asked, paper, go
- **Education:** students, teacher, teachers, child, children, student, school, education, schools, learning, parents, tests, test, program, teaching, behavior, skills, reading, team, information
- **Healthcare:** patient, doctor, health, hospital, medical, dr, patients, nurse, disease, doctors, team, care, office, nursing, drugs, medicine, services, dental, diseases, help

"Illness", "biology", "psychology" and "healthcare" are the topics the learning materials involved. "Education" topic is about the education environment where the chat happened. "Outdoor" and "people/social" are off-task topics.

To get an idea about whether or not the topic scores were related to the learning gain, we aggregated the scores by person and computed the correlation between the total topic score and the learning gain for each topic. We were only interested in looking at the students with larger potential to learn, so we removed the data with pretest score greater than or equal to 0.5, leaving 624 students out of 844. The results (Table 1) showed that all topics were significantly correlated to learning gain. It doesn't seem to be great, because that seems to suggest that, whatever topic a student talked about, more a student talked, larger gain the student obtained. The real reason is that in the aggregation, all topic scores were summed up. Therefore, all topic scores were influenced by the chat length. So the correlation in Table 2 basically showed the chat length effect.

Table 2. Correlation between total topic scores and learning gain (N=624, pretest<0.5)

Topic	*Post-test*	*Pretest*	*Gain*
Illness	.183**	.116**	.132**
Outdoors	.216**	.133**	.154**
Biology	.159**	.125**	.105**
Psychology	.182**	.096*	.140**
People/Social	.115**	.022	.107**
Education	.175**	.118**	.121**
Healthcare	.157**	.130**	.097*

To remove the chat length effect, the simplest way is to divide all scores by the number of words (terms) in each chat. However, in this study, to be consistent with subsequent analysis, we normalized the topic scores to topic proportion scores by dividing each topic score for each utterance by the sum of all seven topic scores of the same utterance.

The results (Table 3) showed that the topic "people/social" had a significant negative correlation to learning gain. Others were not significant but were in the direction we would expect. "Illness", "biology", "psychology" and "healthcare" were positively correlated with gain scores, while "outdoors" and "people/social" topics were negatively correlated with gains scores. We observed almost no correlation for the "Education" topic. This seems to indicate that the aggregated topic scores have limited power in predicting learning. Therefore, we used ENA to examine the connections or association of these topics in the students discourse to

develop a predictive model of learning gains based on the use of these topics.

Table 3. Correlation between normalized topic proportion scores and learning gain (N=624, pretest<0.5)

Topic	*Post-test*	*Pretest*	*Gain*
Illness	.099*	0.077	0.067
Outdoors	-0.063	-0.043	-0.044
Biology	.085*	0.054	0.063
Psychology	0.067	0.019	0.058
People/Social	-.127**	-0.076	-.083*
Education	0.027	0.056	-0.002
Healthcare	0.073	.096*	0.027

5. EPISTEMIC NETWORK ANALYSIS

ENA measures the connections between elements in data and represents them in dynamic network models. ENA creates these network models in a metric space that enables the comparison of networks in terms of (a) difference graph that highlights how the weighted connections of one network differ from another; and (b) statistics that summarize the weighted structure of network connections, enabling comparisons of many networks at once.

ENA was originally developed to model cognitive networks involved in complex thinking. These cognitive networks represent associations between knowledge, skills, habits of mind of individual learners or groups of learners. In this study, we used ENA to construct network models. For each individual student, we constructed an ENA network using the selected seven topic scores for each utterance the student contributed to the group.

5.1 Process

While the process of creating ENA models is described in more detail elsewhere (e.g. [11; 17-19]), we will briefly describe how ENA models are created based on topic modeling. Here we defined network nodes as the seven topics identified from the topic model. We defined the connections between nodes, or edges, as the strength of the co-occurrence of topics within a moving stanza window (MSW) of size 5 [19]. To model connections between topics we used the products of the topic scores summed across all chats in the MSW. That is, for each topic, the topic scores are summed across all 5 chats in the MSW. Then ENA computed the product of the summed topic loadings for each pair topics to measure the strength of their co-occurrence. For example, if the sum of the topics scores across five chats was 0.5 for "illness", 0.3 for "psychology", and 0.2 for "healthcare", these scores would result in three co-occurrences, "illness-psychology", "illness-healthcare", and "psychology-healthcare", with scores of 0.15, 0.1, and 0.06, respectively.

Next ENA created adjacency matrices for each student that quantified the co-occurrences of topics within the students' discourse in the context of their chat group. Subsequently, the adjacency matrices were then treated as vectors in a high dimensional space, where each dimension corresponds to co-occurrence of a pair of topics. The vectors were then normalized to unit vectors. Notice that the normalization removed the effect of chat length embedded in the topic scores. A singular value decomposition (SVD) was then performed for dimensional reduction. ENA then projected a vector for each student into a low dimensional space that maximizes the variance explained in the data. Finally, the nodes of the networks, which in this case correspond to the seven selected topics generated from TASA corpus, were placed in the low dimensional space. The topic nodes were placed using an optimization algorithm such that the overall distances between centroids (centers of the mass of the networks) and the corresponding projected student locations was minimized. A critical feature of ENA is that these node placements are fixed, that is, the nodes of each network are in the same place for all units in the analysis. This fixing of the location of the nodes allows for meaningful comparisons between networks in terms of their connection patterns which allow us to interpret the metric space. As a result, ENA produced two coordinated representations: (1) the location of each student in a projected metric space, in which all units of analysis included in the model were located, and (2) weighted network graphs for each student, which explained why the student was positioned where it was in the space.

ENA also allows us to compare the mean network graphs and mean position in ENA space between different groups of students. In this study, we only considered the students with high potential to learn, i.e., the 624 students with pretest score < 0.5 (50% correct). Among these students, we compared the networks of low learning gain students (gain<-0.1, N=194) with the networks of high learning gain students (gain>0.43, N=105). We compared these groups using difference network graph, which was formed by subtracting the edge weights of the mean discourse network for the low gain group students from the mean discourse network from the high gain group. This difference network graph shows us which topic connections are stronger for each group. In addition, we conducted a *t*-test to test the difference between group means.

5.2 Results

Figure 3 shows mean discourse networks for students with low gain scores (left, red), students with high gain scores (right, blue), and a difference network graph (center) that shows how the discourse patterns of each group differs. Students with low gains had stronger connections between the "people/social" topic and all other topics except for "illness". More importantly, the connection that was the strongest for low gain students compared to high gain students was between "people/social" and "outdoors". Students with high gain scores made stronger connections between the topics of "illness", "psychology", "healthcare", "biology", and "education".

Table 4. Comparison of centroids between low gain and high gain students, $p = 0.047, t = 2.00$

	N	***Mean***	***SD***
High gain	105	0.033	0.220
Low gain	194	-0.048	0.322

Figure 4 shows centroids, or the centers of mass, of individual students' discourse networks and their means with low gain score students in red and high gain score students in blue. The differences between these two groups were significant on the x dimensions (see table 4). This means that the differences we saw in figure 2 and described above are statistically significant. In other words, the high learning gain students' discourse was more towards the right side of the ENA space and the low learning gain students' discourse was more towards the left side. That indicates that the discourse of students with high learning gains made more connections between on-task topics ("illness", "psychology", "healthcare", "biology", and "education"), while the discourse of

low gain students made more connections between off-task topics ("people/social" and "outdoors").

6. DISCUSSION

ENA makes it possible to visualize the chat dynamics to help researchers gain deeper understanding of what is going on in a collaborative learning environment. Differences in what topics students connect in discourse can predict learning outcomes. Previous use of ENA has relied on human coded data or use of regular expressions to classify data. Utilizing topic modeling can lead to fully automated ENA, making it more accessible to a wider group of researchers and allows ENA to be used with more and larger data sets.

The fact that the epistemic network predicts learning validates further application of ENA. For example, the turn by turn chat dynamics can be plotted as trajectories in the 2-D space, where the topics are placed. Investigating the trajectory patterns and their relationship to learning or socio-affective components are interesting future research directions.

We used a general topic model in this study. Many studies in the literature used LDA for topic modeling on relatively small corpora. This causes two problems. 1) LDA topic models built upon small corpora are not reliable, because LDA requires large number documents with relatively large size for each document. Inadequate corpus can result in misleading results. 2) Using a topic model that is not common would result in arbitrary interpretation. For example, the representation of "illness" from different corpus could be very different. Therefore, it is hard to compare the claims made to "illness" across different studies. Using a reliable, common topic models will set up a common language for different studies.

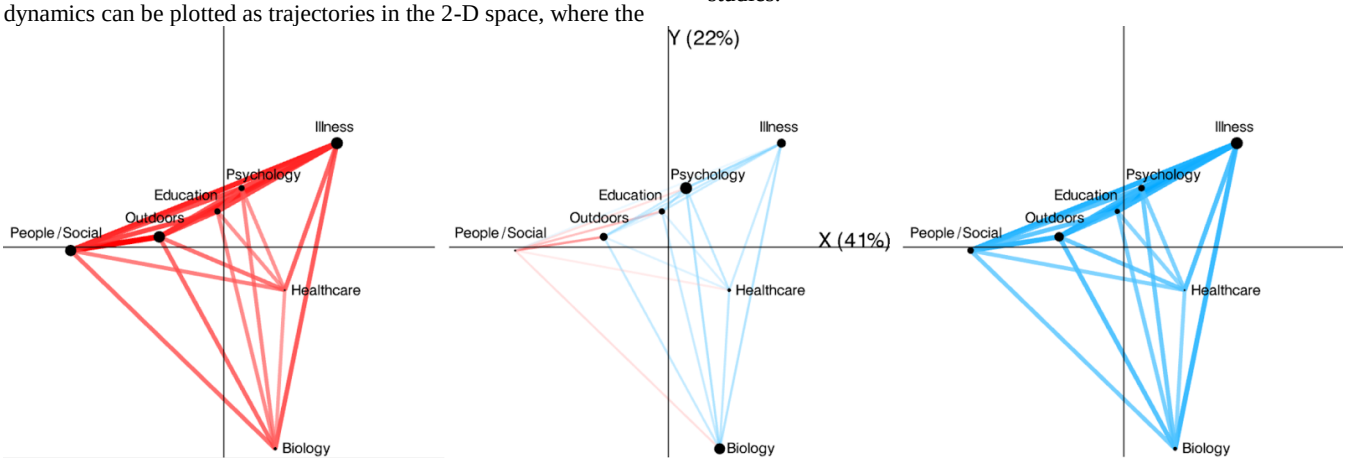


Figure 3: Mean discourse networks for students with low gain scores (left, red), students with high gain scores (right, blue), and a difference network graph (center).

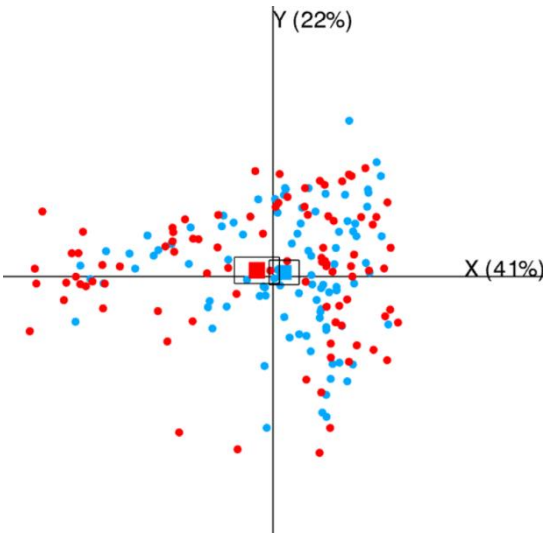


Figure 4: Discourse network centroids low gain score students red, high gain score students blue.

Topic scores for documents are usually inferred from topic models. While for longer documents, the topic scores can be used in many applications (e.g., text clustering [1]), the inferred topic proportion scores won't be useful for analyzing chats if we need to treat each utterance as a unit of analysis. It is not useful because chat utterances are too short. The statistical inference algorithm contains a high degree of randomness for short documents. As an extreme example, an utterance with a single word, would result in inferred topic proportion scores with "1" on one topic and "0" on others. The problem is that, this "1" was assigned to a topic with certain degree of uncertainty. That is, the topic this "1" was assigned to could be any topic. While aggregated analysis may not be sensitive to such uncertainty, detailed utterance by utterance analysis would suffer from it.

Our method of computing topic scores is based on the topic probability distribution over each word. We treat the topic distribution of each word as a vector. When computing the topic score, the simple sum of all word vectors gives scores to all topics. As we have pointed out, the summation algorithm will have a length effect. Therefore, when such topic scores are used, removing length effects through normalization is necessary. In this article, we did not use weighted sum as suggested in Cai et al. [4]. Comparing the effect of different weighting is beyond the scope of this paper.

When a general topic model is used, selecting topics relevant to the specific analysis becomes important. Our approach was to look at the total scores of utterances and find the "hot" topics by sorting the total topic scores. In our study, we had a quickly decreasing curve that helped us to select topics. We believe this would be the case for most studies using a model containing far more topics than the topics contained in the target data.

Although our study started with topic modeling to capture the "what" in the chats, the association networks constructed in the epistemic network analysis actually turned the "what" into a "how": how the topics in the chats associated with each other. This is conceptually similar to the cohesion features Dowell [7] and Cade [3] used.

Topic modeling emphasizes content words. When a topic model is built, stop words are usually removed. An interesting question is, what if we do the opposite: keep stop words and remove content words? Pennebaker (e.g., [13]) laid foundational work in this direction. The LIWC tool Pennebaker and his colleagues created provides over a hundred text measures by counting non-content words. LIWC measures could provide different features to epistemic network analysis and reveal different aspects of the chat dynamics.

7. ACKNOWLEDGMENTS

The research on was supported by the National Science Foundation (DRK-12-0918409, DRK-12 1418288), the Institute of Education Sciences (R305C120001), Army Research Lab (W911INF-12-2-0030), and the Office of Naval Research (N00014-12-C-0643; N00014-16-C-3027). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF, IES, or DoD. The Tutoring Research Group (TRG) is an interdisciplinary research team comprised of researchers from psychology, computer science, and other departments at University of Memphis (visit http://www.autotutor.org).

8. REFERENCES

[1] Alghamdi, R. and Alfalqi, K. 2015. A Survey of Topic Modeling in Text Mining. *IJACSA) International Journal of Advanced Computer Science and Applications*. 6, 1 (2015), 147–153.

[2] Blei, D.M., Edu, B.B., Ng, A.Y., Edu, A.S., Jordan, M.I. and Edu, J.B. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*. 3, (2003), 993–1022.

[3] Cade, W.L., Dowell, N.M.M. and Pennebaker, J. 2014. Modeling Student Socioaffective Responses to Group Interactions in a Collaborative Online Chat Environment. *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*. 2, 21 (2014), 399–400.

[4] Cai, Z., Li, H., Graesser, A.C. and Hu, X. 2016. Can Word Probabilities from LDA be Simply Added up to Represent Documents? *Proceedings of the 9th International Conference on Educational Data Mining*. (2016), 577–578.

[5] von Davier, A.A. and Halpin, P.F. 2013. Collaborative Problem-Solving and the Assessment of Cognitive Skills: Psychometric Considerations. *ETS Research Report Series*. December (2013), 36 p.

[6] Dillenbourg, P. and Traum, D. 2006. Sharing Solutions: Persistence and Grounding in Multimodal Collaborative Problem Solving. *The Journal of the Learning Sciences*. 15, 1 (2006), 121–151.

[7] Dowell, N., Cade, W., Tausczik, Y., Pennebaker, J., and Graesser, A. 2014. What Works: Creating Adaptive and Intelligent Systems for Collaborative Learning Support. *Springer International Publishing Switzerland*. (2014), 124–133.

[8] Dowell, N.M.M., Skrypnyk, S., Joksimović, S., Graesser, A., Dawson, S., Gašević, D., Hennis, T. a., Vries, P. De and Kovanović, V. 2015. Modeling Learners ' Social Centrality and Performance through Language and Discourse. *Educational Data Mining - EDM'15* (2015), 250–257.

[9] Fiore, S.M., Rosen, M. a., Smith-Jentsch, K. a., Salas, E., Letsky, M. and Warner, N. 2010. Toward an understanding of macrocognition in teams: predicting processes in complex collaborative contexts. *Human factors*. 52, 2 (2010), 203–224.

[10] Graesser, A.C., McNamara, D.S., Louwerse, M.M. and Cai, Z. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*. 36, 2 (2004), 193–202.

[11] Li, H., Samei, B., Olney, A., Graesser, A. and Shaffer, D. 2014. Question Classification in an Epistemic Game. *International Conference on Intelligent Tutoring Systems*. (2014).

[12] Pennebaker, J.W., Boyd, R.L., Jordan, K. and Blackburn, K. 2015. The Development and Psychometric Properties of LIWC2015. *Austin, TX: University of Texas at Austin*. (2015).

[13] Pennebaker, J.W., Chung, C.K., Frazee, J. and Lavergne, G.M. 2014. When Small Words Foretell Academic Success : The Case of College Admissions Essays. (2014), 1–10.

[14] Rosen, Y. 2014. Assessing Collaborative Problem Solving Through Computer Agent Technologies. *Encyclopedia of information science and technology*. 9, November (2014), 94–102.

[15] Sawyer, R.K. 2014. The new science of learning. *The Cambridge Handbook of the Learning Sciences*. 1–18.

[16] Scholand, A.J., Tausczik, Y.R. and Pennebaker, J.W. 2010. Assessing group interaction with social language network analysis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 6007 LNCS, (2010), 248–255.

[17] Shaffer, D.W. 2006. Epistemic frames for epistemic games. *Computers and Education*. 46, 3 (2006), 223–234.

[18] Shaffer, D.W., Hatfield, D., Svarovsky, G.N., Nash, P., Nulty, A., Bagley, E., Frank, K., Rupp, A.A. and Mislevy, R.J. 2009. Epistemic Network Analysis: A Prototype for 21st-Century Assessment of Learning. *International Journal of Learning and Media*. 1, 2 (2009), 33–53.

[19] Siebert-Evenstone, A.L., Arastoopour, G., Collier, W., Swiecki, Z., Ruis, A.R. and Shaffer, D.W. 2016. In search of conversational grain size: Modeling semantic structure using moving stanza windows. *International Conference of the Learning Sciences*. (2016).

[20] Slavin, R.E. 1995. Cooperative Learning: Theory, Research and Practice (2nd Ed.). *The Nature of Learning*. (1995), 208.

[21] Tuulos, V.H. and Tirri, H. 2004. Combining Topic Models and Social Networks for Chat Data Mining. *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*. October (2004), 206–

213.

[22] Whitepaper, A.R. 2014. What Happens When We Learn Together. (2014).

[23] Yousef, A.M.F., Chatti, M.A., Schroeder, U., Wosnitza, M. and Jakobs, H. 2014. A Review of the State-of-the-Art. *Proceedings of the 6th International Conference on Computer Supported Education - CSEDU2014*. (2014), 9–20.

[24] Wang Z., Qiu B., Bai, W., Chuan, S. and Le, Y. 2014. Collapsed Gibbs Sampling for Latent Dirichlet Allocation on Spark. *JMLR: Workshop and Conference Proceedings*. 2004 (2014), 17–28.

Towards Closing the Loop: Bridging Machine-induced Pedagogical Policies to Learning Theories

Guojing Zhou, Jianxun Wang, Collin F. Lynch, Min Chi
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
{gzhou3,jwang75,cflynch,mchi}@ncsu.edu

ABSTRACT

In this study, we applied decision trees (DT) to extract a compact set of pedagogical decision-making rules from an original *full* set of 3,702 Reinforcement Learning (RL)-induced rules, referred to as the DT-RL rules and Full-RL rules respectively. We then evaluated the effectiveness of the two rule sets against a baseline Random condition in which the tutor made random yet reasonable decisions. We explored two types of trees (weighted and unweighted) as well as two pruning strategies (pre- and post-pruning). We found that post-pruned weighted trees produced the best results with 529 DT-RL rules. The empirical evaluation was conducted in a classroom study using an existing Intelligent Tutoring System (ITS) named Pyrenees. 153 students were randomly assigned to three conditions. The procedure was the same for all students with domain content and required steps strictly controlled. The only substantive differences between the three conditions were the policy: (Full-RL vs. DT-RL vs. Random). Our result showed that as expected the machine induced policies (Full-RL and DT-RL) are significantly more effective than the random policy; more importantly, no significant difference was found between the Full-RL and DT-RL policies though the number of DT-RL rules is less than 15% of the number of the Full-RL rules and the former group also took significantly less time than the latter.

1. INTRODUCTION

Intelligent Tutoring Systems (ITSs) are interactive e-learning environments that support students' learning by providing instruction, scaffolded practice, and on-demand help. The system's behaviors can be viewed as a sequential decision-making process where at each step the system chooses an appropriate action from a set of options. *Pedagogical strategies* are the policies used to decide what action to take next in the face of alternatives. Each system decision will affect the user's subsequent actions and performance. Its impact on outcomes cannot always be immediately observed and the effectiveness of each decision depends upon the effectiveness of subsequent actions. Ideally, an effective learning environment will adapt its decisions to users' specific needs [1, 11]. However, there is no existing well-established theory on how to make these system decisions effectively. Generally speaking, prior research on pedagogical policies can be divided into two general categories: top-down or *theory-driven*, and bottom-up or *data-driven*.

In theory-driven approaches, ITSs employ hand-coded pedagogical rules that seek to implement existing cognitive or learning theories [1, 10, 17]. While existing learning literature gives helpful guidance on the design of pedagogical rules, such guidance is often too general to implement as effective immediate decisions. For example, the aptitude-treatment interaction (ATI) theory states that instructors should match their interventions to the aptitude of the learner [5]. While the principle behind this theory is understandable, it is not clear how to implement that rule for each decision. How do we represent learner's aptitude for each equation, how exact should be the system's adaptation, and so on.

Data-driven approaches, on the other hand, derive pedagogical policies directly from prior data. Here the policies specify the pedagogical decisions at a detailed level. Reinforcement Learning (RL), which we use here, is one popular approach that is able to derive pedagogical policies directly from student-system interaction logs. These policies are defined as a set of state-action mapping rules, which give the best decision to take in each state. The states are typically represented as sets of features and the actions are pedagogical actions such as presenting a worked example (WE) or requiring the student to solve problems (PS). When the system presents a worked example, the students will be given a detailed example showing a complete expert solution for the problem or the best step to take given their current solution state. In Problem Solving, by contrast, students are tasked with solving a problem using the ITS or with completing an individual problem-solving step.

For this project, our original complete RL-induced policy involves the following seven features representing the students' learning process from different perspectives[1].

[1]In the format of: [Feature-Name] (Discretization Procedure): Explanation of the feature.

1. **[nWESincePS]** **(**$0 \rightarrow 0; (0,1] \rightarrow 1; (1,+\infty) \rightarrow 2$**):** The number of worked example (WE) steps received since the last problem solving (PS) step.

2. **[timeInSession]** **(**$[0,2290] \rightarrow 0; (2290,4775] \rightarrow 1; (4775,7939] \rightarrow 2; (7939,+\infty) \rightarrow 3$**):** The total time spent in the current session.

3. **[avgTimeOnStepPS]** **(**$[0,29.01] \rightarrow 0; (29.01, 48.71] \rightarrow 1; (48.71,+\infty) \rightarrow 2$**):** The average amount of time spent on each PS step.

4. **[avgTimeOnStepSessionPS]** **(**$[0,23.51] \rightarrow 0; (23.51,36.56] \rightarrow 1; (36.56,55] \rightarrow 2; (55,+\infty) \rightarrow 3$**):** The average amount of time spent on each PS step in the current session.

5. **[nStepSinceLastWrongKC]** **(**$[0,1] \rightarrow 0; (1,7] \rightarrow 1; (7,25] \rightarrow 2; (25,+\infty) \rightarrow 3$**):** The number of steps received since the last wrong PS step on the current knowledge component (KC).

6. **[nWEStepSinceLastWrong]** **(**$[0,1] \rightarrow 0; (1,4] \rightarrow 1; (4,10] \rightarrow 2; (10,+\infty) \rightarrow 3$**):** The number of WE steps since the last wrong PS step.

7. **[nCorrectPSStepSinceLastWrongKCSession]** **(**$0 \rightarrow 0; (0,3] \rightarrow 1; (3,10] \rightarrow 2; (10,+\infty) \rightarrow 3$**):** The number of correct PS steps since the last wrong PS step on the current KC in the current session.

With this feature set, a state can be represented as a 7-dimensional vector where each element denotes a discretized feature value. Then, the rules can then be represented as:
(0:0:0:0:0:0:0) -> PS
(0:0:0:0:0:0:1) -> PS
(0:0:0:0:0:1:0) -> PS
(0:0:0:0:0:1:1) -> WE
In this study we discretized the features into three-four values producing a seven-feature state. This results in a state space of $3^2 * 4^5 = 9216$, that is 9216 rules in one RL-induced policy. While these types of polices can specify the exact action to take in each case, they are usually too narrow to be aligned to existing learning theories. Each of the rules covers only a very specific case and the relationship between rules is unknown. Thus it is impossible to explain the power of those rules from the perspective of learning theory. The opacity of those induced rules not only hinders us in improving data-driven methodologies when they go wrong, it also prevents us from advancing learning science research more generally. Moreover, it is possible that some of the decisions are environment-specific and may not generalize to other contexts. This in turn prevents translating these induced policies to environments other than the one from which they are induced. Therefore, a general method is needed to shed some light on the extracted detailed data-driven policies.

Decision tree (DT) induction is a robust data mining approach which can be used to extract a compact set of rules from a set of specific examples. It builds a tree-like hierarchical decision-making pattern which represents the knowledge it learned. Each path from root to leaf represents a single rule which may be dealt with separately. Prior studies have shown that DTs can match training examples in most cases, even with relatively small trees. Davidson et al., for example, built a DT for predicting the extinction risk of mammals [6]. Each of the species was described by 11 ecological features (e.g body mass, geographic range and population density) and were labeled with their extinction risk (threatened vs. non-threatened). Their tree contained 20 general rules which covered 4500 training examples, with a decision accuracy over 80%. Additionally, Reinchard et al. built a DT for predicting the invasiveness of woody plants [13]. The resulting DT encoded 15 rules from 235 examples, with a decision accuracy over 76%. Therefore, in our study, we will apply DT to extract general pedagogical decision-making rules from the detailed RL-induced policies.

In short, our primary research question is: *is DT an effective methodology for extracting more general pedagogical rules from the detailed RL-induced pedagogical rules?* In order to investigate this question, we will build DTs using the rules in a RL-induced policy as training examples and empirically evaluate the effectiveness of the extracted set of DT rules by comparing it to the full set of RL-induced rules in a classroom study. The state features in the RL-induced policies are the input features for the DT and the pedagogical actions are the output labels. In our empirical evaluation, we separate the pedagogical decisions from the instructional content, strictly controlling the content so that it is equivalent for all participants by 1) using an ITS which provides equal support for all learners; and 2) focusing on tutorial decisions that cover the same domain content, in this case WE versus PS.

2. BACKGROUND

2.1 Applying RL to ITSs

Beck et al. applied RL to induce pedagogical policies that would minimize the time students take to complete problems on AnimalWatch, an ITS for grade school arithmetic [2]. They trained the model with simulated students. The low cost of generated data allowed them to apply a model-free RL method, Temporal Difference learning. During the test phase, the induced policies were added to AnimalWatch and the new system was empirically compared with the original system. Their results showed that the policy group spent significantly less time per problem than their no-policy peers. Note that their primary goal was to reduce the amount of time per problem, however faster problem-solving does not always result in better learning performance. Nonetheless, their results showed that RL can be successfully applied to induce pedagogical policies for ITSs.

Iglesias et al., on the other hand, focused on applying RL to improve the effectiveness of an Intelligent Educational System that teaches students DataBase Design [8, 9]. They applied another model-free RL algorithm, Q-learning to induce policies that provide students with direct navigation support through the system's content. They used simulated students to induce the policy and empirically evaluated its effectiveness on real students. Their results showed that while the policy led to more effective system usage behaviors from students, the policy students did not outperform the no-policy peers in terms of learning outcomes.

Shen investigated the impact of both immediate and delayed reward functions on RL-induced policies and empirically evaluated the effectiveness of the induced policies within

an Intelligent Tutoring System called Deep Thought [15]. The induced pedagogical policies are used to decide whether the next task should be WE or PS. They found that some learners benefited significantly more from effective pedagogical policies than others.

Finally, Chi et al. applied model-based RL to induce pedagogical policies to improve the effectiveness of an Intelligent Natural Language Tutoring System for college-level physics called Cordillera [4]. The authors collected an exploratory corpus by training human students on an ITS that makes random decisions and then applied RL to induce pedagogical policies from the corpus. They showed that the induced policies were significantly more effective than the prior ones.

In short, prior studies have shown that RL-induced pedagogical policies can improve students' learning or reduce training time. However, all of these studies focused on the effectiveness of the RL-induced policies. None of them considered extracting more general rules from the induced policies.

2.2 Extracting General Rules

In addition to the work of Davidson et al. [6] and Reinchard et al. [13], DTs have been used for other tasks. Vayssiers et al., for example, applied Classification And Regression Trees to predict the presence of 3 species of oak in California [18]. Their training examples were Vegetation Type Map records for 2085 unique locations. Each record consisted of 25 climatic and geographic features as well as 3 labels showing the presence of the species (Quercus agrifolia, Quercus douglasii and Quercus lobata). One DT was induced for each type. The DTs were tested on another dataset which contains the same type of records for 2016 locations. For Quercus agrifolia, the induced tree had 10 leaf nodes and 94.9% of its predictions are correct for the locations that have the presence of this oak (sensitivity) while 86.7% of its predictions are correct for cases without the oak (specificity). For Quercus douglasii, the induced tree had 22 leaf nodes and a sensitivity and specificity of 87% and 79.9% respectively. For Quercus lobata, the tree had 6 leaves but reached a sensitivity of 77% and a specificity of 73.3%.

Thus, prior studies have shown that DT can effectively extract a small set of general decision-making rules from a large set of specific examples. However, all the examples used by these studies were observations of existing phenomena. So far as we know, this work is the only relevant research on the application of DT to extract a compact set of decision-making rules directly from full RL-induced rules and empirically evaluated the two sets of the rules.

2.3 Applying DT to RL

Prior research on incorporating DT with RL has largely focused on seeking a better representation of state space or policy for RL. Boutilier et al [3]. proposed representational and computational techniques for Markov Decision Processes (MDPs) to reduce the size of the state space. They used dynamic Bayesian networks and DTs to represent stochastic actions as well as DTs to represent rewards. Based upon this representation, they then developed algorithms to find conditional optimal policies. Their method was empirically evaluated on several planning problems and they showed significant savings in both time and space for some types of problems. Gupta et al. proposed the Policy Tree algorithm for RL. This algorithm is designed to directly induce a functional representation of the conditional optimal policies as a DT. They evaluated it on a variety of domains and showed that it was able to make splits properly [7].

In short, prior researchers have shown that properly combining DT with RL can result in a large amount of savings in time and space for finding good policies. However, none of these studies directly applied DT on RL-induced policies.

3. INDUCE FULL SET OF RL-POLICY

Previously, researchers have typically used the Markov Decision Process (MDP) [16] framework to model user-system interactions. The central idea behind this approach is to transform the problem of inducing effective pedagogical policies on what action the agent should take to the problem of computing an optimal policy for an MDP.

3.1 Markov Decision Process

An MDP is a mathematical framework for representing an RL task. It is defined by: a tuple $\langle S, A, T, R\rangle$. Where $S = \{S_1, S_2, ..., S_n\}$ denotes the state space; $A = \{A_1, A_2, ..., A_m\}$ represents a set of agent's possible actions; and $T : S \times A \times S \rightarrow [0, 1]$ is a transition probability table, where each element is $T^a_{S_iS_j} = p(S_j|S_i, a)$. This in turn indicates the probability of transiting from state S_i to state S_j by taking an action a while $R : S \times A \times S \rightarrow \mathbb{R}$ assigns rewards to state transitions given actions. The policy is defined as $\pi : S \rightarrow A$, mapping state S into action A with the goal of maximizing the expected reward.

After defining an MDP, we can transfer the student-system interaction dialog into the trajectory which can then be represented as follows:

$$S_1 \xrightarrow{A_1,R_1} S_2 \xrightarrow{A_2,R_2} S_3 \xrightarrow{A_3,R_3} ... \rightarrow S_N$$

Where $S_i \xrightarrow{A_i,R_i} S_{i+1}$ means that the tutor executed action A_i and received reward R_i in state S_i, and then transferred to the next state S_{i+1}. In general, the reward can be divided into two categories, immediate and delayed, where immediate rewards are received during the state transition, and delayed are available after reaching to goal state.

3.2 Training Datasets

Our training dataset was collected from three exploratory studies in which students were trained on an ITS which made random yet reasonable pedagogical decisions. The studies were given as homework assignments during CSC226: Discrete Mathematics, a core CS course offered at NCSU during the Fall 2014, Spring 2015 and Fall 2015 semesters. The dataset contains a total of 149 students' interaction logs. All students used the same ITS, followed the same general procedure, studied the same training materials, and worked through the same training problems. In order to model the students' learning process, we extracted a total of 142 state feature variables, which can be grouped into five categories:

1. **Autonomy (AM):** the amount of work done by the student: such as the number of problems solved so far *PSCount* or the number of hints requested *hintCount*.

2. **Temporal Situation (TS):** the time related information about the work process: such as the average time taken per problem *avgTime*, or the total time spent solving a problem *TotalPSTime*.
3. **Problem Solving (PS):** information about the current problem solving context, such as the difficulty of the current problem *probDiff*, or whether the student changes the difficulty level *NewLevel*.
4. **Performance (PM):** information about the student's performance during problem solving: such as the number of right application of rules *RightApp*.
5. **Student Action (SA):** the statistical measurement of student's behavior: such as the number of non-empty-click actions that students take *actionCount*, or the number of clicks for derivation *AppCount*.

3.3 Inducing RL Policies

In order to apply RL to induce pedagogical policies, we first defined the pedagogical decision-making problem as an MDP. The state representation includes all of the relevant features available at the beginning of each *step*. The actions are WE and PS at the *step* level. The transition tables were calculated on our training dataset, and our reward function includes two types of reward: delayed and immediate. Our most important reward is based on normalized learning gain (NLG) ($\frac{posttest-pretest}{1-pretest}$), which measures the students' learning gains *irrespective of their incoming competence*. This reward was given as a delayed reward as NLG scores can only be calculated after students finish the entire training process. However, Shen et al. [15] showed that giving immediate rewards can lead to the production of more effective policies when compared to delayed rewards. This is known as the credit-assignment problem. The more that we delay success measures from a series of sequential decisions, the more difficult it becomes to identify which of the decision(s) in the sequence are responsible for our final success or failure. Therefore, for the purposes of this study we also assigned immediate rewards based upon the students' performance during training on the system.

The value iteration algorithm was applied to find the optimal policy. This algorithm operates by finding the optimal value for each state $V^*(s)$. The optimal value for a given state is the expected discounted reward that the agent will gain if it starts in s and follows the optimal policy to the goal. Generally speaking, $V^*(s)$ can be obtained by the optimal value function for each state-action pair $Q^*(s,a)$ which is defined as the expected discounted reward the agent will gain if it takes an action a in a state s and follows the optimal policy to the end. The optimal state value $V^*(s)$ and value function $Q^*(s,a)$ can be obtained by iteratively updating $V(s)$ and $Q(s,a)$ via equations 1 and 2 until they converge:

$$Q(s,a) := R(s,a) + \gamma \sum_{s' \in S} p(S_j|S_i,a)V(s') \quad (1)$$

$$V(s) := \max_a Q(s,a) \quad (2)$$

Here, $p(S_j|S_i,a)$ is the estimated transition model T, $R(s,a)$ is the estimated reward model and $0 \leq \gamma \leq 1$ is a discount factor.

To induce effective pedagogical policies, we combined RL with various feature selections including 10 types of correlation-based methods and an ensemble method and capped the maximum number of state feature size to be eight. More details of our feature selection methods are described in [14]. The final resulting RL policy involves seven state features and 3706 rules.

4. EXTRACTING COMPACT DT-RL SETS

In order to extract a more compact set of decision-making rules from the full set of RL-induced rules, we implemented the ID3 algorithm to build DTs [12]. Each rule in the final RL-induced policy was used as a training example. Two types of decision trees were built: unweighted and weighted, as well as two types of pruning strategies were implemented: pre- and post-pruning. Next, we will discuss each of them in turn.

4.1 Unweighted vs. Weighted Tree

The decision to give a WE vs. PS may impact students' learning differently in different situations. We therefore built two types of decision trees: unweighted and weighted. *Unweighted* trees treated each decision equally while *weighted* trees take account of the relative importance of each pedagogical rule. When applying the value iteration algorithm to induce the optimal policy, we generate the optimal value function $Q^*(s,a)$, which gives the expected discounted reward each agent will gain if it takes an action a in a state s and follows the optimal policy to the end. For a given state s, a large difference between the values of $Q(s,\text{“}PS\text{”})$ and $Q(s,\text{“}WE\text{”})$ indicates that it is more important for the ITS to follow the optimal decision in the state s. We therefore used the absolute difference between the Q values for each state s to weight each RL pedagogical rule.

The ID3 algorithm builds a tree recursively from root to leaves. On each iteration of the construction process the algorithm will check the state of the dataset for the current branch. It will then select a test feature for the current node based upon the weighted information gain. The current node will then be expanded by adding branches to it, each of which represents a possible value for the selected feature. The data will be partitioned over the branches according to the value of the test feature. The selected feature cannot be used again by its children. Weighted information gain is defined by the difference between the weighted entropy of the examples before it is selected and after they are separated by feature value. The weighted entropy of a node can be calculated by equation 3

$$H(G) = -\sum_{i=1}^{J} p(i|G) log_2 p(i|G) \quad (3)$$

J is the total number of output label classes. In our case, it is the number of pedagogical actions (WE or PS) which is 2 . $p(i|G)$ is the weighted frequency defined by the equation: $p(i|G) = \frac{\sum_{x \in i} w_x}{\sum_{y \in G} w_y}$. $\sum_{x \in i} w_x$ is the total weight of the examples which are in node G and which belong to class i. And $\sum_{y \in G} w_y$ is the total weights of examples in node G.

The information gain of spliting the current set of training examples using feature F can be calculated by equation 4:

$$IG(F,G) = H(G) - \sum_{j=1}^{k} p(t_j|G)H(t_j) \quad (4)$$

$p(t_j|G)$ is the weighted frequency of the examples in node G: $p(t_j|G) = \frac{\sum_{x_F=t, x\in G} w_x}{\sum_{y\in G} w_y}$. $\sum_{x_F=t, x\in G} w_x$ is the total weights of examples in nodes G whose value of feature F is j and $\sum_{y\in G} w_y$ is the total weight of examples in nodes G.

4.2 Pre-Pruning and Post-Pruning

To control the size of rules induced by DT, we examined two types of pruning strategy: pre- and post-pruning. The pre-pruning is conducted during the process of building the tree and it used the information gain to determine whether to expand or to terminate. Only nodes with an information gain greater than a threshold times its depth: $IG(F,G) \geq \theta \times D_G$ will be expanded and others will be made as a leaf. θ is a fixed threshold and D_G is the depth of node G.

Post-Pruning is conducted after the whole decision tree is built and it used the error rate as the pruning measure. The error rate before a node is expanded is defined as: $e_G = \frac{\sum_{i\in I} w_i}{|G|}$. I is the set of the decisions incorrectly classified by node G and $|G|$ is the total number of examples in the node G. The error rate after a node is expanded is defined as: $e_C = \frac{\sum_{c\in C}\sum_{j\in I_c} w_i}{|G|}$. C is the set of children nodes of G after it is expanded and I_c is the set of the decisions incorrectly classified by the node c. In post-pruning, if the difference of a node's error rate from before to after split is less than a threshold, the node will be pruned by removing all of its branches to make it a leaf node.

4.3 The Compact Set of DT-RL Rules

In order to induce a compact set of DT-RL rules, we applied the DTs to the full set of 3706 RL-induced rules. The induced unweighted and weighted DTs without pruning has 2527 and 2456 rules (leaf nodes) respectively. Thus, without pruning, DTs are already able to extract a smaller set of rules: it reduced the total number of rules by over 1000.

Figure 1 shows the relationship between the number of leaf nodes (x-axis) and the inverted weighted accuracy (y-axis). Weighted accuracy(WA) is the weighted percentage of decisions correctly made, which can be calculated by the equation: $WA = \frac{\sum_{d_i\in T} w_i}{\sum_{d_i} w_i}$. T is the set of correct predictions made by a DT and w_i is the weight of decision i. The inverted weighted accuracy (IWA) is $IWA = WA^{-10}$, *the lower the better*. Since our goal is to find a good balance point between the IWA and the number of leaf nodes, we applied a widely used strategy called the Elbow Method, to select the best tree. As we can see in the figure, the elbows for the two unweighted tree approaches are around 800 and 1700 rules (x-axis) for the pre and post pruning respectively while the elbows for the two weighted tree approaches are around 250 and 500 for the pre and post pruning respectively. So it seems that weighted tree can extract more compact set of rules than the unweighted trees. While the weighted pre-pruning approach has around 250 rules, its IWA is much higher than the weighted post-pruning approach. Therefore, we chose the weighted tree with post-pruning strategy which has the an elbow at about 500 leaf nodes and reasonable IWA.

To further justify our DT choice, Table 1 shows the relationship between the pruning thresholds, WA and the number

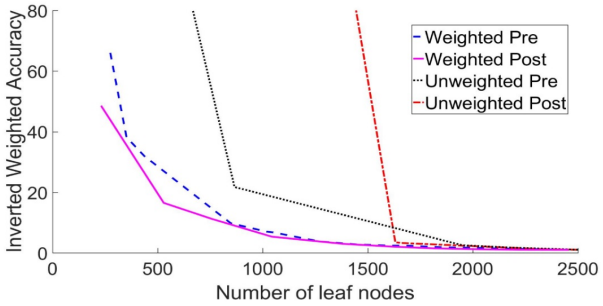


Figure 1: Leaf Nodes - Accuracy

of leaf nodes for the weighted tree with post-pruning. Table 1 shows that the tree with the closest number of leaves to 500 is the 529 one. It can be obtained by apply a pruning threshold of 0.8 and the result tree has a weighted accuracy of 0.76. The rules in the resulted tree will be the rules used in the DT-RL condition.

In short, we applied DT on RL-induced pedagogical policies to extract a more compact set of decision-making rules. The effectiveness of the original full set and the compact set of policies were empirically compared against a baseline policy which makes random yet reasonable decisions: PS vs. WE. Thus, we have three conditions:

1. Full-RL: the full set of 3706 RL-induced rules.
2. DT-RL: the compact set of 529 DT-induced RL rules.
3. Random: the random yet reasonable policy.

5. EMPIRICAL EXPERIMENT

Participants: This study was conducted in the undergraduate Discrete Mathematics course at the Department of Computer Science at NC State University in the Fall of 2016. 153 students participated in this study, which was given as their *final* homework assignment.

Conditions: Students in the study were assigned to three conditions via balanced random assignment based upon their course section and performance on the class mid-term exam. Since the primary goal of this work is to examine the effectiveness of the two RL based policies, we assigned more students to the Full-RL and DT-RL conditions than in the random condition. The final group sizes were: $N = 61$ (Full-RL), $N = 51$ (DT-RL), and $N = 41$ (Random).

Due to preparations for exams and length of the experiment, 126 students completed the experiment. 5 students were excluded from the subsequent analysis due to perfect pretest scores, working in group or gaming the system during the training. The remaining 121 students were distributed as follows: $N = 45$ for Full-RL; $N = 41$ for RL-DT; $N = 35$ for Random. We performed a χ^2 test of the relationship between students' condition and their rate of completion and found no significant difference among the conditions: $\chi^2(2) = 0.955, p = 0.620$.

Probability Tutor: Pyrenees is a web-based ITS for probability. It covers 10 major principles of probability, such as the Complement Theorem and Bayes' Rule. Pyrenees

Table 1: Weighted DT with Post-pruning

Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
WA	1.00	0.99	0.98	0.96	0.93	0.89	0.85	0.79	0.76	0.68
leaves	2456	2217	2029	1809	1608	1383	1043	758	529	231

provides step-by-step instruction and immediate feedback. Pyrenees can also provide on-demand hints prompting the student with what they should do next. As with other systems, help in Pyrenees is provided via a sequence of increasingly specific hints. The last hint in the sequence, the bottom-out hint, tells the student exactly what to do. For the purposes of this study we incorporated three distinct pedagogical decision modes into Pyrenees to match the three conditions.

Procedure: In this experiment, students were required to complete 4 phases: 1) pre-training, 2) pre-test, 3) training on Pyrenees, and 4) post-test. During the pre-training phase, all students studied the domain principles through a probability textbook, reviewed some examples, and solved certain training problems. The students then took a pre-test which contained 14 problems. The textbook was not available at this phase and students were not given feedback on their answers, nor were they allowed to go back to earlier questions. This was also true of the post-test.

During phase 3, students in all three conditions received the same 12 rather complicated problems in the same order on Pyrenees. Each main domain principle was applied at least twice. The minimal number of steps needed to solve each training problem ranged from 20 to 50. These steps included defining variables, applying principles, and solving equations. The number of domain principles required to solve each problem ranged from 3 to 11. All of the students could access the corresponding pre-training textbook during this phase. Each step in the problems could have been provided as either a WE or PS based upon the condition policy. Finally, all of the students completed a post-test with 20 problems. 14 of the problems were isomorphic to the pre-test given in phase 2. The remaining six were non-isomorphic complicated problems.

Grading Criteria: The test problems required students to derive an answer by writing and solving one or more equations. We used three scoring rubrics: binary, partial credit, and one-point-per-principle. Under the binary rubric, a solution was worth 1 point if it was completely correct or 0 if not. Under the partial credit rubric, each problem score was defined by the proportion of correct principle applications evident in the solution. A student who correctly applied 4 of 5 possible principles would get a score of 0.8. The one-point-per-principle rubric in turn gave a point for each correct principle application. All of the tests were graded in a double-blind manner by a single experienced grader. The results presented below are based upon the partial-credit rubric but the same results hold for the other two. For comparison purposes, all test scores were normalized to the range of [0,1].

6. EMPIRICAL RESULTS

Since both the Full-RL and DT-RL policies are based on an RL-induced policy, we combined the two conditions together as the *Induced group* to evaluate the effectiveness the RL-induced policy. The evaluation was conducted by comparing the Induced group with the baseline *Random condition* on learning performance and training time. Moreover, in order to further discover to what extent the compact policy retained the power of the full policy, we compared the Full-RL and DT-RL conditions on the same measures. Next, we will discuss each of the comparisons in turn.

6.1 Induced vs. Random

We measured Students' incoming competence via the pre-test scores collected before training took place. Table 2 shows a comparison between the Induced group and the Random group in terms of learning performance. The parenthesized values following the group names in row 1 denote the number of students in each group. The second row in this table shows the pre-test scores. The last column shows the pairwise t-test results. Pairwise t-tests on students' pre-test scores show that there is no significant difference between the two groups: $t(119) = -0.346$, $p = 0.730$, $d = 0.069$. Thus, despite attrition, the two groups remained balanced in terms of incoming competence. Next, we will compare the two groups in terms of learning performance in the post-test and training time.

Rows 2 - 4 in Table 2 show a comparison of the pre-test, isomorphic post-test (14 isomorphic questions), and adjusted post-test scores between the two groups along with the mean and SD for each. In order to examine the students' improvement through training on Pyrenees, we compared their scores on the pre-test and isomorphic post-test questions. A repeated measures analysis using test type (pre-test and isomorphic post-test) as factors and test score as the dependent measure showed a main effect for test type: $F(1, 119) = 98.75$, $p < 0.0001$. Further comparisons on group by group basis showed that on the isomorphic questions, both groups scored significantly higher in the post-test than in the pre-test: $F(1, 85) = 81.30$, $p < 0.0001$ for Induced and $F(1, 34) = 18.30$, $p = 0.0001$ for Random respectively. This suggests that the basic practice and problems, domain exposure, and interactivity of our ITS might help students to learn even when pedagogical decisions are made randomly.

In order to investigate the effectiveness of the induced policies, we compared students' overall learning performance, which was evaluated by their adjusted post-test scores, between the two groups. A one-way ANCOVA analysis was conducted on their overall post-test scores (20 questions), using the pretest scores as a covariate to factor out the influence of their incoming competence. The result shows a significant main effect: $F(1, 118) = 4.628$, $p = 0.033$. That is, the Induced group significantly outperformed the Random group on adjusted post-test scores, which is shown in

Table 2: Induced vs. Random

Cond	*Induced*(86)	*Random*(35)	T-test Result
Pre	.686(.194)	.699(.171)	$t(119) = -0.346$, $p = 0.730$, $d = 0.069$
Iso Post	.851(.155)	.812(.195)	$t(119) = 1.141$, $p = 0.256$, $d = 0.229$
Adjusted Post	.751(.144)	.689(.138)	$t(119) = 2.162$, $p = 0.033$, $d = 0.433$
Time	105.87(34.30)	111.18(27.33)	$t(119) = -0.815$, $p = 0.417$, $d = 0.163$
WE steps	205.74(62.73)	189.46(11.39)	$t(119) = 1.522$, $p = 0.131$, $d = 0.305$
PS steps	173.69(61.14)	190.26(10.28)	$t(119) = -1.591$, $p = 0.114$, $d = 0.319$
WE pct(%)	54.16(16.35)	49.89(2.78)	$t(119) = 1.532$, $p = 0.128$, $d = 0.307$

the fourth row of Table 2. Therefore, the results showed that the induced policies are significantly more effective than the random policy.

The fifth row in Table 2 shows the average amount of total training time (in minutes) students spent on our ITS for each group. Pairwise t-test showed no significant difference in training time between the two groups: $t(119) = -0.815$, $p = 0.417$, $d = 0.163$. The results suggest that when compared to the random policy, the induced policies generally do not have a significant different impact on students' training time.

The last three rows in Table 2 show the number of WE and PS steps given as well as the percentage of WE steps received by the Induced and the Random group. Pairwise t-tests showed that there is no significant difference between the two groups on these three measures.

6.2 Full-RL vs. DT-RL

We then performed the same comparison between the Full-RL and DT-RL conditions in order to examine the effectiveness of the DT-extracted compact policy. The second row in Table 3 shows the pre-test scores for each condition. A pairwise t-test on the scores shows no significant difference between the two conditions: $t(84) = -0.168$, $p = 0.867$, $d = 0.036$. Thus the two conditions were balanced in terms of incoming competence.

The pre-test, isomorphic post-test and adjusted post-test scores are shown in rows 2 - 4 of Table 3. A repeated measures analysis using test type (pre-test and isomorphic post-test) as factors and test score as dependent measure showed a main effect for test type: $F(1, 85) = 81.30$, $p < 0.0001$. Further comparisons on group by group basis showed that both conditions scored significantly higher in isomorphic post-test than in pre-test: $F(1, 44) = 42.16$, $p < 0.0001$ for Full-RL and $F(1, 40) = 39.16$, $p < 0.0001$ for DT-RL. These results suggest that the students can effectively learn from Pyrenees with the full and compact policies.

In order to discover to what degree the compact policy retained the effectiveness of the full policy, we compared the post-test scores between the two conditions. The results of a pairwise t-test showed no significant different between them on isomorphic post-test: $t(84) = 0.505$, $p = 0.615$, $d = 0.109$. We also conducted an ANCOVA analysis on the overall post-test scores using the pretest scores as a covariate and still found no significant different between the two conditions: $F(1, 83) = 0.348$, $p = 0.557$. In short, while on post-test scores, the DT-RL condition scored slightly lower than the Full-RL condition, the difference is not significant.

The fifth row of Table 3 shows the average amount of time students spent on training. As the row shows, the Full-RL condition spent significantly more time than the DT-RL condition: $t(84) = 3.829$, $p = 0.0002$, $d = 0.827$. Thus the Full-RL and DT-RL policies have significant different impact upon the students' training time.

The last three rows of Table 3 show the number of WE and PS steps given and the percentage of WE steps received by the Full-RL and the DT-RL condition. Pairwise t-tests showed that comparing to the DT-RL condition, the Full-RL condition received significantly fewer WE steps: $t(84) = -4.952$, $p < 0.0001$, $d = 1.069$; received a lower percentage of WE steps: $t(84) = -4.955$, $p < 0.0001$, $d = 1.070$; and completed more PS steps: $t(84) = 4.999$, $p < 0.0001$, $d = 1.079$. These results suggest that the pedagogical decisions made by the compact and full policies are substantively different.

7. DISCUSSION

In this study, we applied DT to extract a compact set of pedagogical rules from the full set of RL-induced rules and empirically evaluated the effectiveness of two sets of rules in a classroom study. Our goal was to shed some light on the RL-induced policies and we think this is only the first step towards narrowing the gap and building a bridge between machine-induced pedagogical policies and learning theories.

In order to find the best DT, we explored two types of tree: unweighted and weighted; and for each of them, we conducted two types of pruning strategy: pre- and post-pruning. After comparing the performance among them, we selected the weighted tree with the post-pruning strategy to perform the extraction of general decision-making rules. The RL-induced policy contains 3706 specific rules, and the compact DT-RL consisted of 529 rules with a weighted decision accuracy of 76%.

In our empirical experiment, we were able to strictly control the domain content and thus to isolate the impact of *pedagogy* from *content*. Based on this isolation, we compared students' performance with the Full-RL policy, the DT-RL policy and the baseline random policy. Our results showed that students in all three conditions learned significantly after training on Pyrenees, this suggests that the basic training of the ITS is effective, even when the pedagogical decisions are made randomly. To evaluate the effectiveness of the two machine induced policies (Full-RL policy and DT-RL policy), we combined the Full-RL and DT-RL condition as the Induced group and compared its learning performance with the Random group. Our results showed that the Induced

Table 3: Full-RL vs. DT-RL

Cond	Full-RL(45)	DT-RL (41)	T-test Result
Pre	.683(.205)	.690(.184)	$t(84) = -0.168,\ p = 0.867,\ d = 0.036$
Iso Post	.859(.145)	.842(.168)	$t(84) = 0.505,\ p = 0.615,\ d = 0.109$
Adjusted Post	.757(.144)	.739(.145)	$t(84) = 0.594,\ p = 0.554,\ d = 0.128$
Time	118.42(35.000)	92.10(27.95)	$t(84) = 3.829,\ p = 0.0002,\ d = 0.827$
WE steps	177.44(48.86)	236.80(62.03)	$t(84) = -4.952,\ p < 0.0001,\ d = 1.069$
PS steps	201.47(47.22)	143.20(60.57)	$t(84) = 4.999,\ p < 0.0001,\ d = 1.079$
WE pct(%)	46.77(12.78)	62.26(16.13)	$t(84) = -4.955,\ p < 0.0001,\ d = 1.070$

group significantly outperform the Random group. These results suggest that the machine induced policies are indeed more effective than the random policy.

Finally, in order to examine to what extent the compact DT-RL policy retained the power of the full RL-induced policy, we compared the learning performance of the Full-RL and the DT-RL conditions. Our results suggest that while some of the power was lost in the general rules extraction, the relative performance difference between the Full-RL and the DT-RL condition is not significant. In addition, our results on the pedagogical decisions made in training revealed that the compact DT-RL policy selected significant more WE than the Full-RL policy. This suggests that the two sets of policies indeed made materially different decisions. However, since the weighted DT took account of the importance of each rule, the DT-RL policy aims to retain maximal decision effectiveness from the Full-RL policy while the size of the former is less than 15% of the size of the Full-RL rules. In the future, we will apply existing learning theories to the decision-making process generated by decision tree to find a theoretical basis for the DT-induced general pedagogical decision-making rules.

8. ACKNOWLEDGEMENTS

This research was supported by the NSF Grant #1432156: "Educational Data Mining for Individualized Instruction in STEM Learning Environments" and #1651909: "Improving Adaptive Decision Making in Interactive Learning Environments".

9. REFERENCES

[1] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207, 1995.

[2] J. Beck, B. P. Woolf, and C. R. Beal. Advisor: A machine learning architecture for intelligent tutor construction. *AAAI/IAAI*, 2000:552–557, 2000.

[3] C. Boutilier, R. Dearden, and M. Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial intelligence*, 121(1):49–107, 2000.

[4] M. Chi, K. VanLehn, D. Litman, and P. Jordan. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 21(1-2):137–180, 2011.

[5] L. J. Cronbach and R. E. Snow. *Aptitudes and instructional methods: A handbook for research on interactions.* Irvington, 1977.

[6] A. D. Davidson and et al. Multiple ecological pathways to extinction in mammals. *Proceedings of the National Academy of Sciences*, 106(26):10702–10705, 2009.

[7] U. D. Gupta, E. Talvitie, and M. Bowling. Policy tree: Adaptive representation for policy gradient. In *AAAI*, pages 2547–2553, 2015.

[8] A. Iglesias, P. Martínez, R. Aler, and F. Fernández. Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning. *Applied Intelligence*, 31(1):89–106, 2009.

[9] A. Iglesias, P. Martínez, R. Aler, and F. Fernández. Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. *Knowledge-Based Systems*, 22(4):266–270, 2009.

[10] K. R. Koedinger and et al. Intelligent tutoring goes to school in the big city. *IJAIED*, 8(1):30–43, 1997.

[11] P. Phobun and J. Vicheanpanya. Adaptive intelligent tutoring systems for e-learning systems. *Procedia-Social and Behavioral Sciences*, 2(2):4064–4069, 2010.

[12] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[13] S. H. Reichard and C. W. Hamilton. Predicting invasions of woody plants introduced into north america. *Conservation Biology*, 11(1):193–203, 1997.

[14] S. Shen and M. Chi. Aim low: Correlation-based feature selection for model-based reinforcement learning. *EDM*, 2016.

[15] S. Shen and M. Chi. Reinforcement learning: the sooner the better, or the later the better? In *UMAP*, pages 37–44. ACM, 2016.

[16] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[17] K. Vanlehn. The behavior of tutoring systems. *IJAIED*, 16(3):227–265, 2006.

[18] M. P. Vayssières, R. E. Plant, and B. H. Allen-Diaz. Classification trees: An alternative non-parametric approach for predicting species distributions. *Journal of vegetation science*, 11(5):679–694, 2000.

On the Influence on Learning of Student Compliance with Prompts Fostering Self-Regulated Learning

Sébastien Lallé
University of British Columbia
2366 Main Mall
Vancouver, BC V6T1Z4, Canada
lalles@cs.ubc.ca

Cristina Conati
University of British Columbia
2366 Main Mall
Vancouver, BC V6T1Z4, Canada
conati@cs.ubc.ca

Roger Azevedo
North Carolina State University
106 Caldwell Hall
Raleigh, NC 27695-8101, USA
razeved@ncsu.edu

Nicholas Mudrick
North Carolina State University
106 Caldwell Hall
Raleigh, NC 27695-8101, USA
nvmudric@ncsu.edu

Michelle Taub
North Carolina State University
106 Caldwell Hall
Raleigh, NC 27695-8101, USA
mtaub@ncsu.edu

ABSTRACT

In this paper, we investigate the relationship between students' learning gains and their compliance with prompts fostering self-regulated learning (SRL) during interaction with MetaTutor, a hypermedia-based intelligent tutoring systems (ITS). When possible, we evaluate compliance from student explicit answers on whether they want to follow the prompts, When such answers are not available, *we mine* several student behaviors related to prompt compliance. These behaviors are derived from students' eye-tracking and interaction data (e.g., time spent on a learning page, number of gaze fixations on that page). Our results reveal that compliance with some, but not all SRL prompts provided by MetaTutor do influence learning. These results contribute to gain a better understanding of how students benefit from SRL prompts, and provides insights on how to further improve their effectiveness. For instance, prompts that do improve learning when followed could be the focus of adaptation designed to foster compliance for those students who would disregard them otherwise. Conversely, prompts that do not improve learning when followed could be improved based on further investigations to understand the reason for their lack of effectiveness

Keywords

Intelligent tutoring systems; Self-regulated learning; Scaffolding; Compliance with prompts; Learning gains; Eye tracking; Linear regression; Hypermedia

1. INTRODUCTION

There is extensive evidence that the effectiveness of Intelligent Tutoring Systems (ITS) is influenced by how well students can regulate their learning, e.g., [13, 22]. Current research has shown that scaffolding self-regulated learning (SRL) strategies such as setting learning goals or assessing progress through the learning content can improve learning outcomes with an ITS, e.g., [1, 10, 22]. In particular, one of the most common approaches to scaffold SRL is to deliver *prompts* designed to guide students in applying specific SRL strategies as needed [22]. Previous work has focused on assessing the general effectiveness of such SRL prompts, for instance by comparing learning outcomes of students working with versions of the same ITS with and without the prompts. (e.g., [1, 19, 21]). Other work has investigated the extent to which students comply with the overall set of prompts generated by an ITS [16, 21]. However, there has been no reported study on the relationship between compliance with *specific* SRL prompts and learning outcomes. In this paper, we aim to fill this gap. Specifically, we explore the impact of student compliance with SRL prompts on learning gains with MetaTutor, an ITS designed to scaffold student SRL processes while learning about topics of the human circulatory system [1].

Our results show that student learning is influenced by compliance with some, but not all, of the SRL prompts delivered by MetaTutor. Overall, we found a positive impact on learning for compliance with prompts fostering learning strategies (revising a summary, reviewing notes), or planning processes (setting new learning goals). On the other hand, we found no impact on learning with prompts related to metacognitive monitoring processes (e.g., prompts to stay on or move away from the current page depending on student performance on a quiz on that page). Having information on the efficacy of each specific prompt in a ITS is important to guide further research on how to improve prompts that do not seem to improve learning when students follow them. Furthermore, prompts that foster learning when followed can become the focus of adaptive interventions designed to improve compliance for those students who would disregard these prompts if left to their own device.

The paper also provides initial insights into prompts design issues that affect how easy it is to evaluate compliance. In MetaTutor, some prompts explicitly asked students whether they wanted to follow the prompt, and then provided suitable affordance to accommodate a positive reply. Compliance with these prompts is easy to assess, but the additional interactions that they require might not always be possible, or might even be intrusive for some students. Other prompts did not require any specific response from the students. Thus, such prompts are in less danger of being intrusive, and provide for a more open-ended interaction. On the other hand, assessing compliance with these prompts is not trivial, because there is no clear definition of what compliance means. For example, one of the MetaTutor prompts asks students to re-read the current MetaTutor content page, but there is no obvious way to map this rather generic suggestion to a specific desired behavior (e.g., spend a specific amount of time on the page, read a specific number of words). We addressed this problem by running linear models to correlate a variety of student behaviors related to prompt compliance with learning. The behaviours we mined are based on both action and eye-tracking data (e.g., time spent on that page, gaze fixations on the content of the page), and our

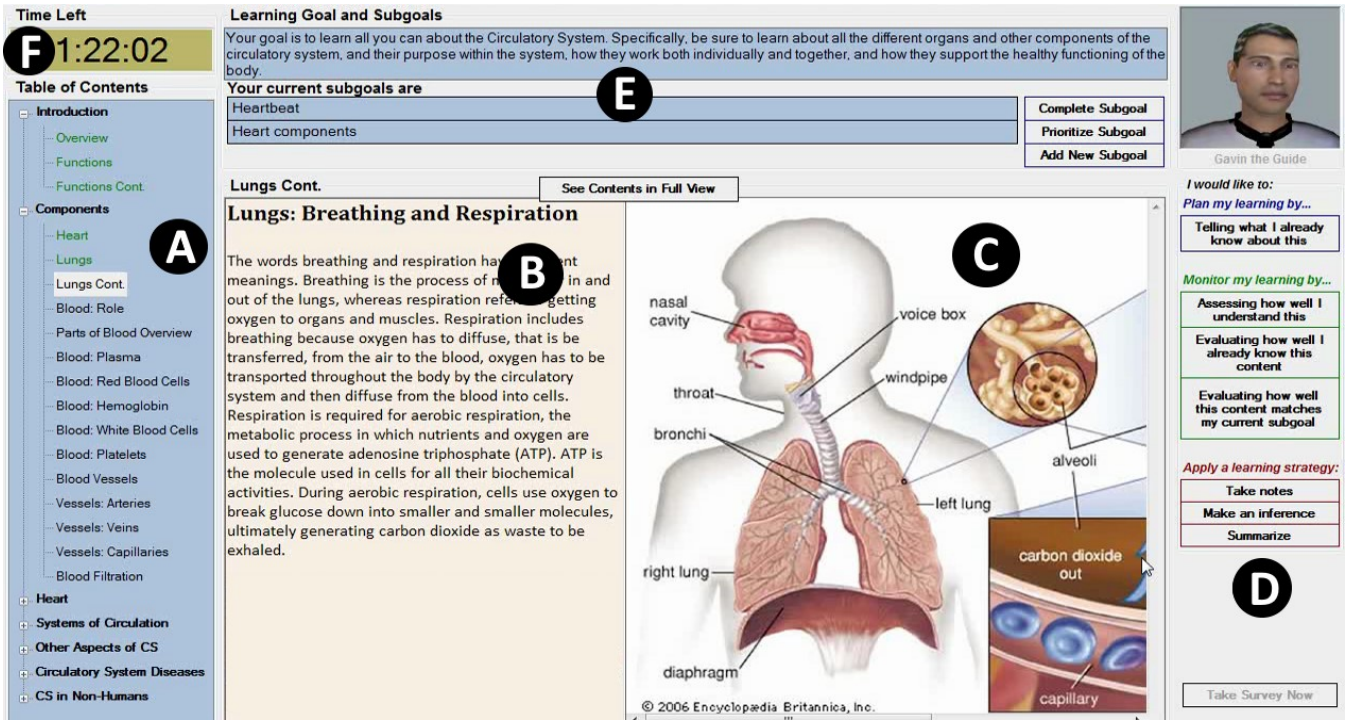


Figure. 1. Screenshot of MetaTutor.

results provide initial evidence that combining these two data sources can help to evaluate compliance. Thus, our findings represent a step toward research on how to evaluate compliance with prompts, both for the type of off line analysis presented in this paper, as well as for the real-time detection of compliance necessary if we want to have ITSs that adaptively help students follow prompts as needed.

The remainder of the paper starts with an overview of related work, followed by a description of MetaTutor and the study that generated the dataset we used for this research. Next, we illustrate how we mined data to evaluate compliance with MetaTutor's prompts, the statistical analysis we conducted, and our results.

2. RELATED WORK

There has been extensive work on assessing the effectiveness of scaffolding designed to support learning with ITSs. Scaffolding can include *prompts* or *hints* (i.e., interventions that guide the student in the right direction), *feedback* (evaluation of students answers, behavior or strategies), or *demonstration* (e.g., worked examples showing expert behavior) [22, 23]. Such scaffolding can be *domain-specific* to support the acquisition of domain-specific knowledge, or targeting domain-independent, meta-cognitive learning processes such as processes for self-regulated learning (SRL). There is extensive evidence that both domain-specific scaffolding (e.g., [3, 12, 18, 20]) and meta-cognitive scaffolding (e.g., [2, 10, 11, 21]) can improve the effectiveness of ITS. For example, domain-specific hints that explain how to solve the current problem step have been shown to improve skill acquisition in a variety of domains such as mathematics [20] and reading [3, 12]. At the meta-cognitive level, Roll et al. [21] tracked suboptimal help-seeking patterns (e.g., overuse of help) to deliver prompts and feedback on how to effectively use help. Prompts and feedback designed to help construct self-explanations during reading [10] or solving scientific problems [11] have been found to positively influence learning. Azevedo et al. [2] showed that SRL prompts and feedback effectively foster efficient use of SRL strategies while learning about biology.

Research has also examined student compliance with SRL prompts in ITS [5, 16]. Kardan and Conati [16] examined the benefit of providing a variety of prompts designed to help students progress within an interactive learning simulation. Overall they found that students largely complied with the prompts and that providing these prompts improved learning gains. However, they did not explore whether and how compliance with specific prompts influence learning outcomes, and which prompts are the most effective. Bouchet et al. [5] adapted the frequency of prompt delivery in MetaTutor based on whether students previously complied with prompts of the same type. However, their analysis uncovered no influence of such adaptive prompting strategy on learning gains. We extend the aforementioned work on prompt compliance by showing how learning gains are impacted by compliance with some, but not all SRL prompts in MetaTutor. Furthermore, whereas previous solely used interaction data to evaluate compliance, we also leverage eye-tracking data when compliance cannot be inferred directly from students' answers or actions (e.g., compliance with the prompts of reading a text further).

Eye-tracking has been used in ITS to model a variety of students traits and behavior, e.g., emotions [14], learning outcomes [15], metacognitive behavior [7], or mind wandering [4]. Eye tracking has also been used to capture students attention to prompts [6, 8] and to pedagogical agents [17]. Conati et al. [6] leveraged gaze data to detect whether students processed domain-specific textual prompts in an educational game for math, and found that reading the prompts more extensively improved game performance. Lallé et al. [17] used gaze data to capture student visual attention to pedagogical agents in MetaTutor, and found that student learning gains are significantly influenced by specific metrics for visual attention (fixation rate, longest fixation). Eye-tracking has also

been used to add real-time adaptive prompts to Guru, an agent-based ITS for learning biology [9]. In that work, audible prompts designed to reorient student attention towards the screen were triggered if a student had not looked at the screen for more than 5s while Guru was providing scaffolding. This research showed that this gaze-reactive feedback can improve learning with Guru. In our work, we mine eye-tracking data to evaluate compliance with specific SRL prompts, and examine whether and how compliance with such SRL prompts influences learning gains.

3. METATUTOR

MetaTutor [1] is a hypermedia-based ITS containing multiple pages of content about the circulatory system, as well as mechanisms to help students self-regulating their learning with the assistance of multiple speaking pedagogical agents (PAs). When working with MetaTutor, students are given the overall goal of learning as much as they can about the human circulatory system. The main interface of MetaTutor (see Fig. 1) includes a table of contents (Fig. 1A), the text of the current content page (Fig.1B), a miniature image allowing the student to display a diagram along with the text (Fig. 1C), the current goals and subgoals to learn about (Fig. 1E), a timer indicating how much time remains in the learning session (Fig. 1F), and an SRL palette (Fig. 1D). This palette is designed to scaffold students self-regulatory processes by providing buttons they can select to initiate specific SRL activities (e.g., making a summary, taking a quiz, setting subgoals). Further SRL scaffolding is provided by three PAs in the form of *feedback* on student performance on these SRL activities (e.g., performance on quiz or on the quality of their summaries), as well as *prompts* designed to guide these activities as needed. The PAs deliver these prompts based on student behavior (e.g., time spent on page, number of pages visited).

Specifically, *Pam the Planner* prompts planning processes primarily at the beginning of the learning session by suggesting to add a new subgoal and, if needed, which one to choose (e.g., path of blood flow, heart components). *Mary the Monitor* scaffolds students' metacognitive monitoring processes by making them take quizzes on the target material when they appear to be ready for them. Based on quiz outcomes, Mary prompts students to evaluate the relevance of the current content and subgoal to their knowledge, and suggests how to move through the available material and sub goals accordingly. *Sam the Strategizer* prompts students to apply the learning strategies consisting of summarizing the content studied so far or reviewing notes they have taken on the content[1].

All PAs provide audible assistance through the use of a text-to-speech engine (Nuance). The PAs are visually rendered using Haptek virtual characters, which generate idle movements when the PAs are not speaking (subtle, gradual head and eye movements), as well as lip movements during speech.

4. USER STUDY

The data used for the analysis presented in this paper were collected via a user study designed to gain a general understanding of how students learn with MetaTutor [1]. The study included the collection of a variety of multi-channel trace data (e.g., eye tracking, log files, physiological sensors). In this paper, we focus on using interaction and eye-tracking data to track compliance with the SRL prompts provided by MetaTutor, and study the relationship among compliance with the prompts and learning gains.

Twenty-eight college students participated in the study, which consisted of two sessions conducted on separate days. During the first session, lasting approximately 30-60 minutes, students were administered several questionnaires, including a 30-item pretest to assess their knowledge of the circulatory system. During the second session lasting approximately three hours, students first underwent a calibration phase with the eye tracker (SMI RED 250) as well as a training session on MetaTutor. Each student was then given 90 minutes to interact with the system. Finally, students completed a posttest analogous to the pretest, followed by a series of questionnaires about their experience with MetaTutor.

5. DATA ANALYSIS

5.1 Evaluating Compliance with Prompts

In our analysis we categorize prompts into two types based on how compliance can be evaluated. The first type includes prompts for which compliance can be explicitly assessed from students subsequent responses (*explicit compliance prompts*); the second type includes prompts for which compliance needs to be inferred by mining a variety of behaviors (*inferred compliance prompts*).

Explicit compliance prompts are those that:

- Require students to answer "yes" or "no" (using a dialogue panel that becomes active at the bottom of the display). If students answers yes, the only action they can perform in the MetaTutor interface is the one they agreed upon (e.g., adding a specific subgoal suggested by the agent, making or revising a summary, moving to a previously added subgoal or staying on the current one)[2].
- Require students to take a specific action within a specific time frame (i.e., open the diagram while they are on the current page, and review notes by the end of the learning session).

Table 1 lists the explicit compliance prompts considered in this analysis.

Inferred compliance prompts are those for which the PAs do not force students to provide an explicit answer. Specifically, after the agent utters one of these prompts, the student simply clicks on "continue" in the same dialogue panel, and can either ignore the prompted action, or comply at some point. These prompts (listed in Table 2) include all prompts related to staying on or moving away from the current page, as well as initiating the action of adding a new subgoal.

5.2 Statistical Analysis

Our analysis aims to investigate if and how compliance with MetaTutor's SRL prompts influence learning. The variable we

[1] More details about the design of the agents can be found in [1].

[2] For the "stay on current subgoal" prompt, students are not forced to comply after answering "yes", but we have listed it in this category because student are still required to explicitly answer "yes" or "no" to the PAs as for whether they want to follow the prompt or not.

Table 1. List of explicit compliance prompts provided in MetaTutor (grouped by type of prompted SRL processes).

Prompt label	Description	Prompts for
Suggest subgoal	Recommend possible subgoals to learn about while the students is adding new subgoal.	Planning processes
Moving to next subgoal	Recommend moving on to another subgoal when the student did well on a quiz related to the current subgoal.	Metacognitive monitoring processes
Stay on subgoal	Recommend to learn more about the current subgoal when the student did not do well enough on a quiz related to that subgoal.	
Open diagram	Recommend opening the diagram when it is relevant to the current subgoal.	
Summarize	Recommend making a summary of the current page when the student has spent enough time on that page.	Learning strategies
Revise summary	Recommend revising the summary submitted by the student when there are issues with the summary (e.g., the summary is too long or too short).	
Review notes	Recommend reviewing notes taken on the learning content when approaching from the end of the session.	

Table 2. List of inferred compliance prompts provided in MetaTutor (grouped by type of prompted SRL processes).

Prompt label	Description	Prompts for
Add subgoal	Recommend adding a new subgoal to learn about when a student has no active subgoal.	Planning processes
Move to next page	Recommend moving on to another page when the student did well on a quiz related to the current page.	Metacognitive monitoring processes
Stay on page	Recommend staying on the current page when the student did not well enough on a quiz related to that page.	

adopted to measure learning in our analysis is *proportional learning gain*, defined as:

$$\frac{posttest\ score\ ratio - pretest\ score\ ratio}{1 - pretestscore\ ratio}$$

Table 3 reports statistics for pre- and post-test scores, as well as for the corresponding learning gains.[3]

Table 3. Descriptive statistics for pretest, posttest, and learning gain.

Measures of learning	M	SD	Median
Pretest	18.6	4.2	19
Posttest	21.4	4	21
Proportional learning gain	15.3	50.2	20

We conducted two separate analyses for explicit and inferred compliance prompts, described next.

Explicit compliance prompts. Since compliance is directly observed in the data for explicit compliance prompts (listed in Table 2), we computed a *compliance rate* for each of these prompts as follow:

$$\frac{Number\ of\ prompts\ followed}{Total\ number\ of\ prompts\ delivered}$$

[3] The increase from pretest to post-test is statistically significant indicating that MetaTutor is overall effective at fostering learning, as further discussed in [1].

Table 4 shows the compliance rate averaged across students for each of the seven explicit compliance prompts in MetaTutor, and the number of prompts delivered.

Table 4. Descriptive statistics of the number of explicit compliance prompts delivered, as well as on compliance rate.

Prompt	Total number of prompts delivered	Compliance rate *Mean (SD)*
Suggest subgoal	60	.90 (.25)
Move next subgoal	25	.85 (.34)
Stay on subgoal	44	.27 (.37)
Open diagram	77	.21 (.32)
Summarize	105	.32 (.41)
Revise summary	59	.76 (.37)
Review notes	28	.46 (.51)

To investigate the impact of compliance with explicit compliance prompts on learning, we ran a multiple linear regression model with *proportional learning gain* as the dependent variable, as well as the *compliance rate* for each of the seven explicit compliance prompts, and the *total number of prompts received* as the factors. For post-hoc analysis we ran pairwise *t*-test comparisons, and *p*-values were adjusted with the Holm-Bonferroni approach to account for multiple comparisons.

Inferred compliance prompts. As stated above, for inferred compliance prompts (listed in Table 5), students are not forced to explicitly accept or ignore the prompt. This means that compliance with those prompts has to be assessed from student behaviors following the prompts. One approach we considered was to make this assessment binary, as we did for explicit compliance prompts, by establishing thresholds for relevant behaviors. For instance, compliance with the prompt to re-read the current page could be assessed to be true if the student stays on the page for a fixed number of seconds after receiving this prompts. However, it

is difficult to fix these thresholds in an informed manner, as they may depend on the student (e.g., on a student's readings speed, existing understanding of the page, etc.), and on the object of the prompt (e.g., on the length or difficulty of the page to be re-read). It is also difficult to decide which specific behaviors should be considered for compliance, as several might be relevant (e.g., time spent on a page, specific attention patterns on a page).

Thus, for the subsequent analysis, we avoided committing to specific thresholds and behaviors, and we opted instead for performing regression analyses to try to relate multiple relevant compliance behaviors to learning.

We started by building *data windows* that capture student data from the delivery of each inferred compliance prompt in Table 2, to the following actions:

- "Moving to another page" for the *move to next page* and *stay on page* prompts;
- "Adding a new subgoal" for the *add new subgoal* prompt.

We used these data windows to derive three behavioral measures related to compliance:

- *Window length*, capturing how long students spent before moving on to another page or adding a new subgoal;
- *Number of fixations*[4] made on MetaTutor's learning content (text and diagram), as captured by eye tracking. We use this measure to understand whether students read the page and/or processed the diagram;
- *Number of SRL strategies* initiated by the student by pressing the corresponding buttons in the SRL palette (see Fig. 1 D).

Higher values of these measures (i.e., long windows, high number of fixations on the page and high number of SRL strategies used) are possible indicators that the student is processing the current page, e.g., the student is thinking about or reading the content (as captured by the length of the data window and number of fixations on the page), or using SRL strategies on the current page. Thus, we hypothesized that higher values of these measures could reveal compliance with *stay on page* prompts, whereas lower values could reveal compliance with prompts instructing students to *move on*. Similarly, because prompts to *add a subgoal* requires moving on from the learning content to actually add a subgoal, we expected a short window, a small number of fixations on the page, and a small number of SRL strategies to indicate compliance.

It should be noted that we could have generated other eye-tracking measures, such as fixation duration on the text or the number of transitions from the text to other components of the MetaTutor's interface. However, because valid eye-tracking data were collected for only 16 students out of the 28 who participated in the study, resulting in a rather small dataset, we focused on the most promising behavioral measures that could be related to compliance, as a proof of concept. Table 5 shows the amount of inferred compliance prompts delivered to those 16 students.

Table 5. Number of inferred prompts delivered.

Prompt	Total number of prompts delivered
Add a subgoal	34
Stay on page	117
Move to next page	326

We leveraged the three aforementioned measures of student behavior to investigate if complying with inferred compliance prompts influences learning, and if so, how. Specifically, for each of the three inferred compliance prompts, we ran a multiple linear regression model with *proportional learning gain* as the dependent variable, as well as the *window length*, *number of SRL strategies performed*, and *number of fixations on the learning content* as the factors. As done for explicit compliance prompts, we used pairwise *t*-test comparisons for post-hoc analysis, and all *p*-values were adjusted with the Holm-Bonferroni approach.

6. RESULTS

We describe below the significant[5] effects found in our analysis, first for explicit compliance prompts, and second for inferred compliance prompts.

6.1 Effects for Explicit Compliance Prompts

Our statistical analysis uncovered significant main effects of *compliance rate* for three explicit compliance prompts:

- *Revise summary* ($F_{1,20} = 6.17$, p=.02, $\eta_p^2 = .15$), shown Fig. 2a.
- *Review notes* ($F_{1,20} = 7.43$, p=.013, $\eta_p^2 = .16$), shown Fig. 2b.
- *Suggest subgoal* ($F_{1,20} = 11.4$, p=.003, $\eta_p^2 = .27$), shown Fig. 2c.

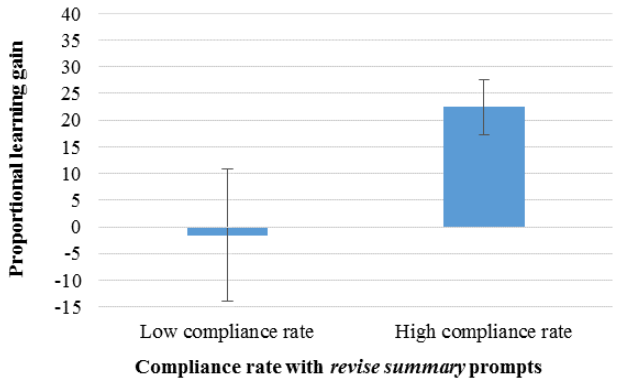
These three main effects and related pairwise comparisons all reveal that students learned more when they complied more with these prompts than when complying less.

These results for *revise summary* and r*eview notes* are consistent with previous findings showing these learning strategies can be beneficial for learning [17, 22, 24], and extend them by showing that prompting these strategies is effective when students comply with the prompts. Notably, we found a significant effect for prompts to *revise summary*, but not for prompts to *summarize*. This indicates that solely prompting to summarize is not enough to improve learning, and that guiding the students through the process of making a good summary is necessary. Results for *suggest subgoal* indicate that recommending a particular learning subgoal is useful, possibly because it is difficult for students to choose good subgoals by themselves.

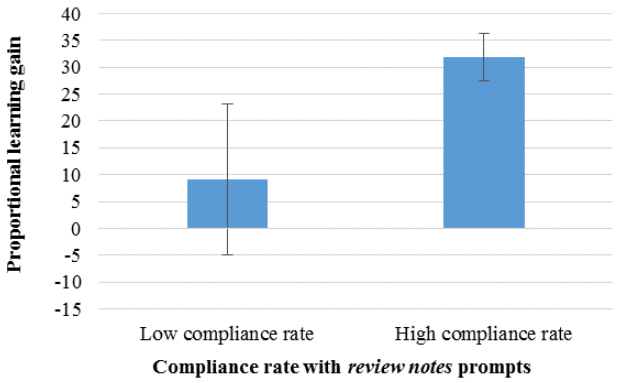
These results suggest to examine ways to improve compliance with prompts to *revise summary, review notes* and *suggest subgoal*, since our analysis reveals that not complying with them hinders learning. For instance, MetaTutor could foster compliance with these prompts by explaining how they can help the students, or conversely force the students to follow these prompts.

[4] Fixation is defined as gaze maintained at one point on the screen for at least 80ms.

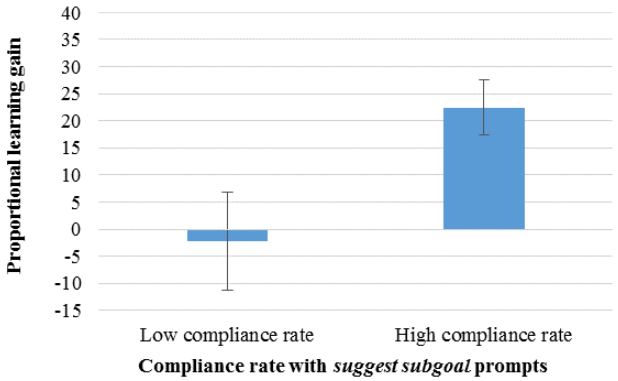
[5] We report statistical significance at the 0.05 level throughout this paper, and effect sizes as small for $\eta_p^2 \geq 0.02$, medium for $\eta_p^2 \geq 0.13$, and large for $\eta_p^2 \geq 0.26$.



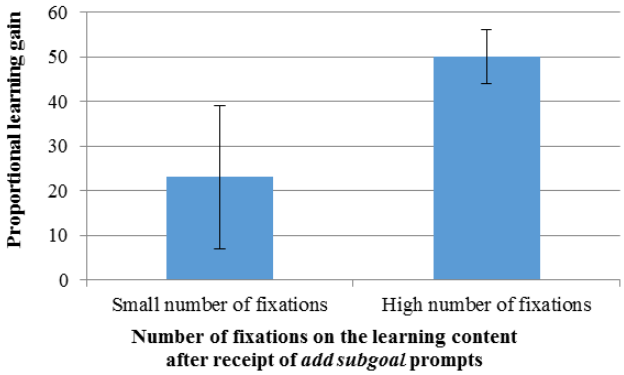
a. Main effect of compliance rate with "revise summary".



b. Main effect of compliance rate with "review notes".



c. Main effect of compliance rate with "suggest subgoal".



d. Main effect of fixation on page after reception of "add subgoal".

Figure 2. Main effects found in this analysis, for explicit compliance prompts (charts a, b, c) and inferred conpliance prompts (chart d). Error bars show 95% confidence interval.

We found no significant effects and small effect sizes (see Appendix A) for the four remaining prompts, namely *summarize*, *stay on subgoal* or *move to next subgoal*, and *open the diagram*. These results indicate it is important to study the effectiveness of SRL prompts individually, to identify those for which compliance does not improve learning. Based on these findings, it is justified to further investigate why complying with these prompts is not beneficial for learning in MetaTutor, and revise the prompts accordingly. For example, it might be due to the nature of the prompts, their timing, their frequency, their wording, and so forth.

6.2 Effects for Inferred Compliance Prompts

We found a main effect of *fixation on learning content* for the "add subgoal" prompts ($F_{1,3} = 13$, $p = .03$, $\eta_p^2 = .29$), shown in Fig. 2d. This effect and related pairwise comparisons reveal that students learned more when they fixate more on the current page than when fixating less. Since students were instructed to add a new subgoal rather than process the current page, this finding suggests that complying with this prompt might not be effective for learning with MetaTutor, possibly because of the timing of this prompt, its frequency or its wording. Although only seven students with valid gaze data received this prompt, the effect size is large, suggesting it is worth conducting further analysis to ascertain whether and why complying with this prompt is not beneficial for learning.

We found no effects and small effect sizes (see Appendix B) for the other inferred compliance prompts, namely *stay on page* and *move to next page*, two prompts related to metacognitive monitoring processes. We cannot make final conclusions on the pedagogical effectiveness on these prompts based on these results, because the dataset is not large and for this reason we did not include in the analysis other features that could indicate compliance (for example other eye-tracking measures such as fixation duration on text or gaze transitions from the text to other components of MetaTutor). However, it should be noted that we also found no effect for the explicit compliance prompts that foster metacognitive monitoring processes (*stay on subgoal*, *move to next subgoal*, and *open the diagram*, see previous section). This lack of effect for all prompts fostering metacognitive monitoring, even when compliance is explicitly assessed, suggests that these prompts are not beneficial for learning with MetaTutor. This could be due to the way these prompts are currently implemented in MetaTutor (e.g., their wording, timing delivery or frequency), or to the nature or the prompts itself. Our results nonetheless justify to run further analysis to ascertain whether (and why) prompts fostering metacognitive monitoring are not effective, and revise them as needed.

7. CONCLUSION

In this research we investigated the relationship between compliance with prompts designed to support the use of self-regulated learning (SRL) processes and learning gains while learning about

the human circulatory system with MetaTutor. We identified two approaches to evaluate compliance to MetaTutor's prompts:

(i) Assess compliance from students' subsequent response to the prompts when students are forced to express compliance (e.g., by answering "yes" or "no" to a prompt);

(ii) Run linear models to examine the influence on learning of a variety of student behaviors related to prompt compliance, when compliance is not elicited by MetaTutor. The behaviors we mined are based on both interface and eye-tracking data (e.g., time spent on that page, gaze fixations on the content of the page).

Our results revealed that student learning gains are influenced by compliance with some, but not all SRL prompts provided by MetaTutor. Specifically, we found a positive influence on learning for prompts that foster learning strategies (*revise a summary* and *review notes*) as well as prompts that recommend setting a specific learning subgoal. Based on these findings, it is worth exploring ways to improve compliance with these prompts. In particular, in future research we plan to examine whether forcing students to comply with these prompts or providing detailed explanations on how the prompted SRL strategies can be useful can improve learning.

We found that compliance with the other MetaTutor's prompts studied in this analysis does not improve learning. This finding reveals that assessing compliance to SRL prompts individually is useful to identify prompts that may not be effective at supporting learning. In particular, we found no results for all prompts related to metacognitive monitoring processes (e.g., staying on/moving away from the current page), suggesting to examine further why complying with these prompts do not influence learning with MetaTutor. For example, it could be due to their timing and frequency, their wording, their nature, and so forth.

In this paper we also addressed the challenge of evaluating compliance with rather open-ended prompts for which there is no clear definition of compliance. Specifically we ran a linear regression analysis to relate relevant compliance behaviors to learning. Such behaviors were derived from a combination of student interaction and eye-tracking data after receipt of a prompt (e.g., time spent and amount of gaze fixations on a page can reveal compliance with prompt to read that page). Preliminary results show that such interaction-based and eye-tracking-based measures can help evaluate compliance. In future research, we plan to investigate further behavioral measures relevant to assessing compliance, such as tracking eye gaze patterns on the different components of MetaTutor as well as transitions between those components.

Lastly, we plan to investigate the possibility of detecting in real time compliance with SRL prompts for which we found a positive effect on learning, using eye-tracking and interaction data. Such real-time detection could inform the design of adaptive prompts to foster compliance for those students who might otherwise disregard these prompts. For instance, adaptive prompts could force students to follow them or explain how the prompted SRL processes can improve learning. Evaluating such adaptive prompts fostering SRL processes would provide further insights on how students comply with and benefit from SRL prompts.

8. ACKNOWLEDGMENTS

This publication is based upon work supported by the National Science Foundation under Grant No. DRL-1431552 and the Social Sciences and Humanities Research Council of Canada. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Social Sciences and Humanities Research Council of Canada.

9. REFERENCES

[1] Azevedo, R., Harley, J., Trevors, G., Duffy, M., Feyzi-Behnagh, R., Bouchet, F. and Landis, R. 2013. Using trace data to examine the complex roles of cognitive, metacognitive, and emotional self-regulatory processes during learning with multi-agent systems. *International handbook of metacognition and learning technologies*. Springer, 427–449.

[2] Azevedo, R., Martin, S.A., Taub, M., Mudrick, N.V., Millar, G.C. and Grafsgaard, J.F. 2016. Are Pedagogical Agents' External Regulation Effective in Fostering Learning with Intelligent Tutoring Systems? *Proceedings of the 13th International Conference on Intelligent Tutoring Systems* (Zagreb, Croatia, 2016). Springer, 197–207.

[3] Beck, J., Chang, K., Mostow, J. and Corbett, A. 2008. Does help help? Introducing the Bayesian Evaluation and Assessment methodology. *Proceedings on the 9th International Conference on Intelligent Tutoring Systems* (Montréal, QC, Canada, 2008). Springer, 383–394.

[4] Bixler, R. and D'Mello, S. 2015. Automatic Gaze-Based Detection of Mind Wandering with Metacognitive Awareness. *Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization* (Dublin, Ireland, 2015). Springer, 31–43.

[5] Bouchet, F., Harley, J.M. and Azevedo, R. 2016. Can Adaptive Pedagogical Agents' Prompting Strategies Improve Students' Learning and Self-Regulation? *Proceedings of the 13th International Conference on Intelligent Tutoring Systems* (Zagreb, Croatia, 2016). Springer, 368–374.

[6] Conati, C., Jaques, N. and Muir, M. 2013. Understanding attention to adaptive hints in educational games: an eye-tracking study. *International Journal of Artificial Intelligence in Education*. 23, 1–4 (2013), 136–161.

[7] Conati, C. and Merten, C. 2007. Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Know.-Based Syst.* 20, 6 (2007), 557–574.

[8] D'Mello, S., Olney, A., Williams, C. and Hays, P. 2012. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of Human-Computer Studies*. 70, 5 (2012), 377–398.

[9] D'Mello, S., Olney, A., Williams, C. and Hays, P. 2012. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of Human-Computer Studies*. 70, 5 (2012), 377–398.

[10] Graesser, A. and McNamara, D. 2010. Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educational Psychologist*. 45, 4 (2010), 234–244.

[11] Hausmann, R.G. and Vanlehn, K. 2007. Explaining self-explaining: A contrast between content and generation. *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (Los Angeles, CA, USA, 2007). Springer, 417–424.

[12] Heiner, C., Beck, J. and Mostow, J. 2004. Improving the help selection policy in a Reading Tutor that listens. *Proceedings of the InSTIL/ICALL Symposium on NLP and Speech Technologies in Advanced Language Learning Systems* (Venice, Italy, 2004), 195–198.

[13] Jacobson, M.J. 2008. A design framework for educational hypermedia systems: Theory, research, and learning emerging scientific conceptual perspectives. *Educational technology research and development*. 56, 1 (2008), 5–28.

[14] Jaques, N., Conati, C., Harley, J.M. and Azevedo, R. 2014. Predicting Affect from Gaze Data during Interaction with an Intelligent Tutoring System. *Proceedings of the 12th International Conference on Intelligent Tutoring Systems* (Honolulu, HI, USA, 2014). Springer, 29–38.

[15] Kardan, S. and Conati, C. 2012. Exploring gaze data for determining user learning with an interactive simulation. *Proceedings of the 20th international conference on User Modeling, Adaptation, and Personalization* (Montréal, QC, Canada, 2012). Springer, 126–138.

[16] Kardan, S. and Conati, C. 2015. Providing adaptive support in an interactive simulation for learning: An experimental evaluation. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, South Korea, 2015). ACM, 3671–3680.

[17] Lallé, S., Taub, M., Mudrick, N.V., Conati, C. and Azevedo, R. 2017. The Impact of Student Individual Differences and Visual Attention to Pedagogical Agents during Learning with MetaTutor. *Proceedings of the 18th International Conference on Artificial Intelligence in Education* (Wuhan, China, 2017). Springer (to appear).

[18] McNamara, D.S., Boonthum, C., Levinstein, I.B. and Millis, K. 2007. Evaluating self-explanations in iSTART: Comparing word-based and LSA algorithms. *Handbook of latent semantic analysis*. Psychology Press. 227–241.

[19] Najar, A.S., Mitrovic, A. and McLaren, B.M. 2014. Adaptive Support versus Alternating Worked Examples and Tutored Problems: Which Leads to Better Learning? *Proceedings of the 22nd International Conference on User Modeling, Adaptation, and Personalization* (Aalborg, Denmark, 2014). Springer, 171–182.

[20] Poitras, E.G. and Lajoie, S.P. 2014. Developing an agent-based adaptive system for scaffolding self-regulated inquiry learning in history education. *Educational Technology Research and Development*. 62, 3 (2014), 335–366.

[21] Ritter, S., Anderson, J.R., Koedinger, K.R. and Corbett, A. 2007. Cognitive Tutor: Applied research in mathematics education. *Psychonomic bulletin & review*. 14, 2 (2007), 249–255.

[22] Roll, I., Aleven, V., McLaren, B.M. and Koedinger, K.R. 2011. Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*. 21, 2 (2011), 267–280.

[23] Roll, I., Wiese, E.S., Long, Y., Aleven, V. and Koedinger, K.R. 2014. Tutoring self-and co-regulation with intelligent tutoring systems to help students acquire better learning skills. *Design Recommendations for Intelligent Tutoring Systems, Volume 2*. U.S. Army Research Laboratory. 169–182.

[24] Shute, V.J. 2008. Focus on formative feedback. *Review of educational research*. 78, 1 (2008), 153–189.

[25] Trevors, G., Duffy, M. and Azevedo, R. 2014. Note-taking within MetaTutor: interactions between an intelligent tutoring system and prior knowledge on note-taking and learning. *Educational Technology Research and Development*. 62, 5 (2014), 507–528.

APPENDIX A

All statistical results for explicit compliance prompts (discussed in Section 6.1). Bold indicates a significant effect.

Prompt	F value	*p*-value	Effect size
Suggest subgoal	**$F_{1,20} = 11.4$**	***p*=.003**	**η_p^2=.27**
Review notes	**$F_{1,20} = 7.43$**	***p*=.013**	**η_p^2 =.16**
Revise summary	**$F_{1,20} = 6.17$**	***p*=.02**	**η_p^2 =.15**
Summarizing	$F_{1,20} = 1.76$	*p*=.20	η_p^2 =.06
Move on subgoal	$F_{1,20} = 0.92$	*p*=.35	η_p^2 =.02
Stay on subgoal	$F_{1,20} = 1.47$	*p*=.24	η_p^2 =.01
Open diagram	$F_{1,20} = 0.71$	*p*=.41	η_p^2 =.08

APPENDIX B

All statistical results for explicit compliance prompts (discussed in Section 6.2). Bold indicates a significant effect.

Prompt	Measure	F value	*p*-value	Effect size
Add subgoal	Window length	$F_{1,3} = .91$	$p = .41$	$\eta_p^2 = .04$
	#fixations on page	**$F_{1,3} = 13$**	**$p = .03$**	**$\eta_p^2 = .29$**
	#SRL strategies	$F_{1,3} = .02$	$p = .90$	$\eta_p^2 = .01$
Move on page	Window length	$F_{1,10} = .00$	$p = .98$	$\eta_p^2 = .00$
	#fixations on page	$F_{1,10} = .03$	$p = .86$	$\eta_p^2 = .00$
	#SRL strategies	$F_{1,10} = .40$	$p = .54$	$\eta_p^2 = .01$
Stay on page	Window length	$F_{1,10} = .34$	$p = .57$	$\eta_p^2 = .01$
	#fixations on page	$F_{1,10} = .07$	$p = .79$	$\eta_p^2 = .03$
	#SRL strategies	$F_{1,10} = .004$	$p = .95$	$\eta_p^2 = .02$

Assessing Computer Literacy of Adults with Low Literacy Skills

Andrew M. Olney
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
aolney@memphis.edu

Dariush Bakhtiari
Department of Educational Psychology, Special Education, and Communication Disorders
Georgia State University
Atlanta, GA 30302
dbakhtiari1@gsu.edu

Daphne Greenberg
Department of Educational Psychology, Special Education, and Communication Disorders
Georgia State University
Atlanta, GA 30302
dgreenberg@gsu.edu

Art Graesser
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
a-graesser@memphis.edu

ABSTRACT

Adaptive learning technologies hold great promise for improving the reading skills of adults with low literacy, but adults with low literacy skills typically have low computer literacy skills. In order to determine whether adults with low literacy skills would be able to use an intelligent tutoring system for reading comprehension, we adapted a 44 task computer literacy assessment and delivered it to 114 adults with reading skills between 3rd and 8th grade levels. This paper presents four analyses on these data. First, we report the pass/fail data natively exported by the assessment for particular computer-based tasks. Second, we undertook a GOMS analysis of each computer-based task, to predict the task completion time for a skilled user, and found that it negatively correlated with proportion correct for each item, $r(42) = -.4,\ p = .01$. Third, we used the GOMS task decomposition to develop a Q-matrix of component computer skills for each task, and using logistic mixed effects models on this matrix identified five component skills highly predictive of the success or failure of an individual on a computer task: function keys, typing, using icons, right clicking, and mouse dragging. And finally, we assessed the predictive value of all component skills using logistic lasso.

Keywords

adult literacy, computer literacy, GOMS, Q-matrix, mixed model, lasso

1. INTRODUCTION

Of adults with the lowest literacy levels, 43% live in poverty, and low literacy costs the U.S. economy $225 billion annually [14]. The need for literacy interventions is matched by the complexity of delivering interventions to this population. Low literacy adults have difficulty attending face to face programs at literacy centers because of work, child care, and transportation [5], and even when these challenges are met, two-thirds of literacy centers have long waiting lists [14]. Adaptive computer-based interventions for literacy hold promise to overcome these challenges. Such interventions can be deployed in homes and local libraries, in addition to literacy centers. However, computer-based interventions raise another question: can adults with low literacy skills use computers well enough to benefit? Several surveys suggest that this might be a problem. The demographics most affected by low literacy are the same demographics least likely to use the Internet (over age 50, making less than $30 thousand a year, and with less than a high school education [1]).

Several decades of research have investigated computer literacy using self-report measures as well as objective tests, i.e. multiple choice, and find that self-report measures tend to exaggerate proficiency while objective tests are more reliable (see [3] for a review). For an adult literacy population, however, multiple-choice tests delivered as print create additional concerns as to whether the questions themselves can be comprehended. Recently a new type of assessment, known as the Northstar Digital Literacy Assessment (the Northstar), has been created that directly measures ability to perform computer tasks [13]. Unlike multiple choice assessments, the Northstar can simulate a computer desktop, use voice prompts to instruct users to perform tasks on that desktop, and then record their mouse clicks and keystrokes to determine if the task has been completed. Almost all of the tasks can be completed without reading by listening to the voice prompt instructions. The few tasks that do involve reading are word recognition tasks rather than sentence reading, e.g. a task to log in may require the user to copy a name and password to the appropriate boxes and so require reading of "Username," "Password," and the corresponding fillers. The Northstar has been adopted as the computer literacy standard for adult basic education in the

state of Minnesota, which further supports its appropriateness for assessing the computer literacy skills of adults with low literacy skills.

The present study investigated the computer literacy skills of adults with low literacy skills for the purpose of developing an intelligent tutoring system for reading comprehension for this population [7]. It includes a set of Northstar items that were collected to cover a range of potential interface and interaction components. In the remainder of the paper we describe the data collection procedure and four analyses performed, including pass/fail frequencies for each task, relation of these frequencies to GOMS-predicted execution times for skilled users, a logistic mixed-model using a Q-matrix decomposition of the tasks into component skills, and a logistic lasso model to assess the predictive value of component skills. From these analyses we identify specific tasks that are problematic for adults with low literacy skills as well as component skills that make it more likely adults with low literacy skills will succeed or fail at a computer-based task.

2. ANALYSIS 1: PROPORTION CORRECT

2.1 Participants

Participants (N = 114) were recruited through adult literacy centers in Atlanta, GA and Toronto, ON, from classes where the reading level was between 3rd and 8th grade. Reading level was determined by the centers using their "business as usual" assessments. Demographic surveys were completed by 90 participants (79% completion rate). Completed surveys indicated that participants were slightly more female than male (55 vs. 35) and that participant age ranged from 17 to 69 ($M = 42.74,\ SD = 13.73$).

2.2 Materials

Forty-four items were selected from four (out of seven) of the Northstar modules available at the time of the study: Basic Computer Skills (21), WWW (13), Windows (6), and Email (4). Task descriptions are given in Table 1. Basic Computer Skills covered such topics as turning a computer on, identifying components of a computer, files and folders, menus, and windows. WWW focused on browser-based activities like searching, search results, browser functionalities, and logging in. Although the Windows module focused on Windows overall, the items selected were fairly generic to any windowed operating system and mostly pertained to desktop applications. Email questions used a webmail interface (browser-based email client) and queried how one would create a new email, send an email, or similar email task. Because Northstar modules are integrated assessments, the Northstar Project compiled the items we selected into a custom assessment for us.

2.3 Procedure

Participants first completed informed consent and then the demographic survey. Both informed consent and demographic survey were read aloud to participants to ensure comprehension. Participants were then asked to sit in front of a computer to take the Northstar assessment. The assessment was delivered in the browser using Adobe Flash. At the start of the assessment, a 3-minute orientation video was played explaining how to answer questions in the assessment. If the participant was confused about what to do, an experimenter was available to answer questions. Each question consisted of an voice prompt defining the task, which was also written at the top of the screen. A replay button was available to repeat the prompt. Participants could select, click, type, drag, etc. on the interface in an attempt to perform the task. If the participant did not know how to complete the task, they could press an "I Don't Know" button, at which point the system scored their attempt as a failure. Attempts were only scored as a success if the participant completed the task in the manner requested in the prompt. The completion of each task initiated the next task until the assessment was complete.

2.4 Results & Discussion

The Northstar records success/failure of each participant on each task, and these data are reported in detail elsewhere [2]. Here we briefly note that the proportion of correct responses for each task is quite wide, ranging from .19 to .98. Tasks in which participants performed particularly well (proportion correct above .80) include identification tasks (e.g. for mouse, keyboard, headphone jack, and websites), turning on a computer or monitor, and common operations like recycling a file, using checkboxes, dragging, scrolling, and using hyperlinks. Tasks in which participants performed poorly (proportion correct below .60) include identification of various keys, double- or right-clicking, typing web addresses, signing into email, and composing email.

The proportion correct results from the Northstar indicate the adults with low literacy skills can power on their device and perform a variety of basic operations. To the extent that these tasks exactly matched tasks that would be performed in a computer-based literacy intervention, like an intelligent tutoring system, this level of results is quite useful. However, for some tasks there is not an exact match, and the implications of the proportion correct results are less clear. For example, difficulties performing tasks using Word, Excel, or webmail may reflect problems with those specific interfaces that may not transfer to other programs. Understanding these more nuanced relationships would require a deeper analysis than is afforded by Northstar's success/failure output.

3. ANALYSIS 2: GOMS MODELING

The purpose of this analysis was to explore whether the success rate of the Northstar tasks could be modeled using GOMS (Goals, Operators, Methods, & Selection rules), a well-known computational technique for modeling expert user performance on a task [10]. GOMS decomposes a particular computer task, e.g. saving a file, into goals and subgoals, perceptual, cognitive, and motor actions in service these goals, methods or sequences of operators that achieve a goal, and selection rules that choose between alternative methods. An important assumption of GOMS is that the users are expert at the computer task in question. Therefore GOMS models of execution time represent the upper bound of performance after a user has learned the interface and practiced it many times. The expert assumption of GOMS is violated in the adult literacy population, making the outcome of this analysis non-obvious. If the GOMS model predictions of execution time were related to our adult's performance, that would provide evidence that GOMS modeling

Table 1: Northstar Tasks

Click on the monitor	Recycle file	Click stop loading
Click on the keyboard	Checkboxes	Select search engines
Click on the system unit	Organize folder options	Google query
Click on the headphone jack	Start menu, lauch program	Google scroll
Click on picture of a mouse	Turn up audio slider	Use hyperlink
Newline key	Mute audio	Maximize window
Caps key	Select browser icons	Minimize window
Shift key	Click on the website	Open Excel
Backspace key	Drag item in browser	Open Word using taskbar
Up arrow	Click on address bar	Close Word
Turn on monitor	Type the web address	Select login and password
Turn on computer	Click homepage button	Choose secure password
Log on to computer	Click browser back button	Sign into email
Double click on Documents	Click browser refresh	Compose email
Right click menu	Click browser forward	

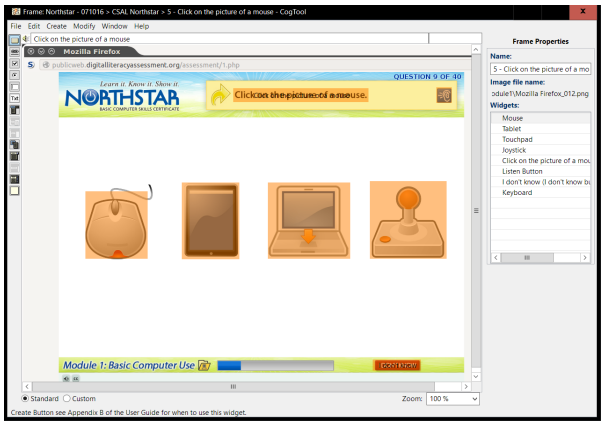


Figure 1: A CogTool annotation of a Northstar task. Annotations appear as semi-transparent orange boxes over the Northstar interface.

has some validity for this population.

3.1 Procedure

The CogTool system was used to perform a GOMS analysis [11, 9]. CogTool allows the easy creation of GOMS models by annotating an existing user interface, and then recording a demonstration of the task against than annotated interface. Figure 1 shows the CogTool interface for the "Click on the mouse" task. For example, when the Northstar task required clicking on an icon, button, or other interface element as in Figure 1, a CogTool button annotation was overlaid on the interface, and then in demonstration mode the modeler would demonstrate the task by clicking on the annotated button. From this demonstration on the annotation, CogTool builds a GOMS model that includes the perceptual, cognitive, and motor tasks required to perform the task. Similar annotations were made for auditory directions, keyboard input, and other kinds of interface actions. Once a task was annotated and demonstrated, a CogTool simulation was run on GOMS model to generate a predicted execution time of expert performance. Annotations, demonstrations, and execution time predictions were performed for all 44 Northstar items used in Analysis 1.

3.2 Results & Discussion

GOMS-predicted execution times for Northstar tasks ranged from 3.0 to 17.1 seconds ($M = 6.88,\ SD = 4.07$). These execution times were significantly negatively correlated with proportion correct, $r(42) = -.40,\ p = .01,\ CI_{95}[-.61, -.10]$, indicating that tasks predicted to take an expert longer to accomplish were more likely to be answered incorrectly by low literacy adults. Tasks that take longer are inherently more complex and require more operations to complete. These results suggest that GOMS has some validity for modeling the performance of adults with low literacy skills even though it was not intended for this purpose. However, by themselves these results convey little additional insight. The GOMS-predicted execution times, generated by CogTool, are still at the task level rather than the component skills required to achieve each task. This is partly because the orientation of CogTool is to produce execution times and partly because of the expert orientation of GOMS. For example, in GOMS the factors involved in clicking a button are the perceptual (size, location) and motor operations involved, but in Northstar, some "buttons" are tapping specific types of knowledge, like identifying hardware, understanding icons, or various keys on a keyboard. The different types of knowledge behind the various CogTool annotations are not represented or considered in the GOMS analysis it provides.

4. ANALYSIS 3: Q-MATRIX & LOGISTIC MIXED MODELS

We would like to understand how the component skills underlying Northstar tasks differentially affect the probability a low literacy adult will perform the task correctly. In educational data mining, component skills are typically modeled using a Q-matrix analysis [4]. In its simplest form, a Q-matrix analysis constructs a problem by skill matrix such that a $cell_{ij}$ in the matrix represents whether $skill_i$ is needed to solve $problem_j$: $cell_{ij} = 1$ if $skill_i$ is needed to solve $problem_j$, and $cell_{ij} = 0$ if $skill_i$ is not needed to solve $problem_j$. Analysis 2 provides a useful guide towards the creation of a Q-matrix for the Northstar tasks, as it has already captured each component action required to perform

Table 2: Component skills coded from GOMS

Component Skill	Probability Correct Given Skill
Checkboxes	.89
Mouse Drag	.86
Hardware Identify	.83
Hardware Function	.78
Complex Scrolling	.74
Browser Functions	.66
Left Click	.64
Use Icons	.61
Double Click	.58
Window Functionality	.56
Program Brands	.55
Desktop Concept	.53
Select Menu	.50
Good Login Info	.50
Login Info	.48
Keyboard Function	.46
Simple Typing	.43
Right Click	.19

each task. What it lacks in some cases, however, is an annotation of the knowledge behind each component action.

4.1 Procedure

The first author recoded the GOMS task annotations with 18 novice-relevant component skills. The coding was done in one pass, and component skills were defined on the fly. Component skills that occurred in only one task were then removed as they offer no predictive utility for other tasks. The appropriateness of the component skills was evaluated by correlating the total number of component skills needed in each task with the GOMS execution time and the proportion correct for the respective task. We used a logistic mixed model to predict the correctness of each participant on each task as a function of the presence of component skills for that task. This analysis addresses the question as to whether there is an effect (main effect) of the presence of component skills on the likelihood that an adult with low literacy skills will be able to perform the task correctly. Using a logistic mixed model in this way has strong similarities to cognitive psychometric models like Diagnostic Classification Models [16] or more specifically a mixed model implementation of linear logistic test models [15].

In the logistic mixed model, random slopes were initially included but failed to converge. Random intercepts for task and participant are theoretically motivated, and backward selection of these effects using Akaike information criterion (AIC) achieved a minimum when these effects were included, indicating that these intercepts should remain in the model. These random intercepts can be considered as per-task difficulty not captured by component skills and per-subject ability, respectively. The initial model that included Left Click was rank deficient, so Left Click, which appears in most tasks, was removed from the final model. Additionally, the total number of component skills in each task (i.e. column sums of the Q-matrix) was initially considered as a predictor of correctness, but was excluded based on extremely high collinearity, having a variance inflation factor of over 40.

4.2 Results & Discussion

The component skills and the conditional probability that a task will be correctly performed if the component skill is present are shown in Figure 2. Total component skills per task was marginally positive correlated with GOMS execution time, $r(42) = .27,\ p = .07,\ CI_{95}[-.02, .53]$, suggesting that tasks with more component skills take longer to perform. Total component skills per task was significantly negatively correlated with proportion correct, $r(42) = -.35,\ p = .02,\ CI_{95}[-.59, -.06]$, indicating that tasks with more component skills are more difficult to perform correctly. The correlation between predicted execution time and proportion correct was not significantly different from the correlation between total component skills and proportion correct, $t(82) = .18,\ p = .86$, indicating that the Q-matrix decomposition of component skills is comparable to the GOMS execution time in terms of its relationship to proportion correctness. Altogether these correlation results provide additional evidence that the Q-matrix decomposition is appropriate.

The logistic mixed model had a marginal R^2 of .18 (fixed effects only) and a conditional R^2 of .47 (including random effects) [12]. We found a positive main effect of Mouse Drag, $\hat{\beta} = 2.06,\ SE = .90,\ p = .02$, such that tasks with a Mouse Drag component were 7.87 times as likely to be answered correctly, $CI_{95}[1.36, 45.50]$, and a marginal main effect of Hardware Identify, $\hat{\beta} = .89,\ SE = .53,\ p = .10$, such that tasks with a Hardware Identify component were 2.44 times as likely to be answered correctly, $CI_{95}[.86, 6.94]$. We found negative main effects for Keyboard Function, $\hat{\beta} = -1.31,\ SE = .51,\ p = .01$, Use Icon $\hat{\beta} = -1.35,\ SE = .55,\ p = .01$, Simple Typing $\hat{\beta} = -1.91,\ SE = .64,\ p = .003$, and Right Click $\hat{\beta} = -3.20,\ SE = 1.34,\ p < .02$, such that tasks with a Keyboard Function component were .27 times as likely to be answered correctly, $CI_{95}[.10, .73]$, tasks with a Use Icon component were .26 times as likely to be answered correctly, $CI_{95}[.09, .75]$, tasks with a Simple Typing component were .15 times as likely to be answered correctly, $CI_{95}[.04, .52]$, and tasks with a Right Click component were .04 times as likely to be answered correctly, $CI_{95}[.00, .56]$.

We found that Mouse Drag was extremely predictive of success. The reason is unclear, but we hypothesize that the frequency of mouse dragging in many computer tasks may have afforded participants the opportunity to become expert in this skill. Mouse dragging has some similarity to swiping on a smartphone or tablet interface, so it may be that expertise with other devices has transferred into the Northstar tasks. Amongst the components that predict failure, perhaps the most intuitive are Keyboard Function and Simple Typing. Typing is a complex skill that takes practice to master. Function keys are difficult in that they don't themselves produce a character, but either operate on a character on the screen (Delete) or work in combination with another key to modify it (Shift). The negative effects associated with Use Icon and Right Click are somewhat surprising. Icons come in many different variations, and so it is possible that the negative Use Icon effect is attributable to a lack of knowledge of specific icons or perhaps to the conventions of icons generally. Right Click is possibly rare and usually brings up a context menu with commands that are often available elsewhere, making it more relevant for power users but perhaps less so to novice users.

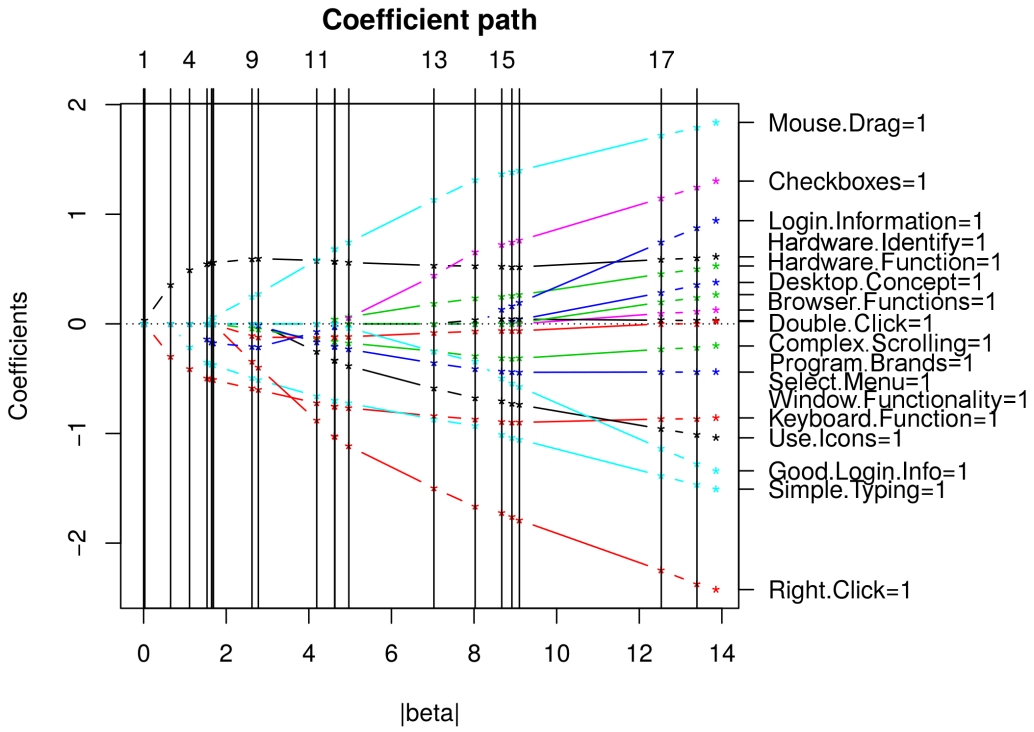


Figure 2: The coefficient path for the lasso model. As the L1 sparsity threshold increases along the x-axis, more coefficients are non-zero.

5. ANALYSIS 4: Q-MATRIX LASSO

Analysis 3 provides a more traditional analysis of significant predictors in our study, but must be interpreted with caution with respect to generalizing to new data. It may be that insignificant predictors in Analysis 3 nevertheless have predictive value on new data. The problems of relying on p-values or criteria like AIC to select variables are well known [8]. To explore the predictive potential of the Q-matrix component skills, we created a lasso model (least absolute shrinkage and selection operator [18]), a form of regression that promotes sparsity (i.e. zero coefficients) and predictive accuracy simultaneously. While not necessarily the best predictive model (cf. gradient boosting [6]), lasso has the advantage of being simple to interpret, and thus our results can guide what variables to use in future models.

5.1 Procedure

A logistic regression base model without random effects was initialized with 17 component skills (Left Click excluded) and submitted to lasso. Because lasso has a free parameter, λ, that controls sparsity of the regression, a lasso analysis varies the level of λ and generates regression coefficient estimates at each level. This sequence of regression coefficients is known as the regularization path. The value of λ that minimized prediction error was estimated using both cross validation and AIC.

5.2 Results & Discussion

The coefficient (regularization) path for the lasso model is shown in Figure 2 and the corresponding AIC curve is shown in Figure 3. In Figure 2, the center line represents coefficients having zero values. As the L1 sparsity threshold ($|beta|$) increases, more coefficients become non-zero. For selecting the optimal λ that minimizes overall prediction error, ten-fold cross validation and AIC yielded congruent results. AIC results are depicted in the curve in Figure 3, which shows that that AIC improves as $|beta|$ increases, coming to a minimum at $|beta| = 13.40$. Accordingly, most coefficients for the optimal lasso model are non-zero.

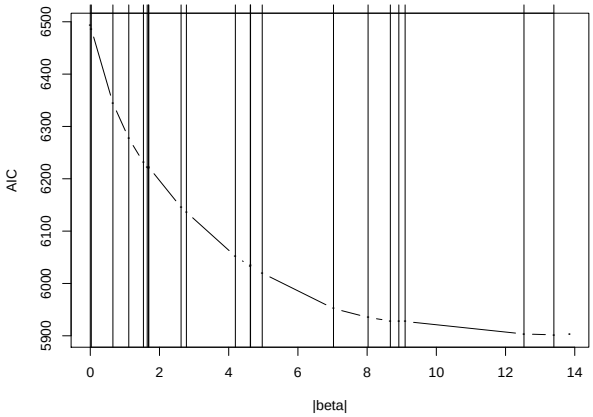


Figure 3: The AIC curve for the lasso model. Lower values of AIC indicate better model fit.

Table 3: Lasso component skill coefficients

Component Skill	$\hat{\beta}$	$exp(\hat{\beta})$
Mouse Drag	1.80	6.02
Checkboxes	1.27	3.55
Login Information	.88	2.41
Hardware Identify	.60	1.82
Hardware Function	.50	1.65
Desktop Concept	.35	1.43
Browser Functions	.24	1.27
Double Click	.11	1.12
Complex Scrolling	.03	1.03
Program Brands	.00	1.00
Select Menu	-.21	.81
Window Functionality	-.44	.64
Keyboard Function	-.86	.42
Use Icons	-1.01	.36
Good Login Info	-1.28	.28
Simple Typing	-1.47	.23
Right Click	-2.39	.09

Table 3 gives the $\hat{\beta}$ coefficients (log odds) for the AIC-optimal model as well as the odds ratio $exp(\hat{\beta})$ for each coefficient. The coefficients converted to odds ratios have the same interpretation as in the logistic mixed model, e.g. tasks with a Mouse Drag component are 6.02 times as likely to be answered correctly as those without. Although the logistic lasso model does not include random intercepts corresponding to task difficulty and subject ability, the magnitudes of coefficients in the logistic lasso are highly comparable to the logistic mixed model. However, the strength of the coefficients in the logistic lasso are weaker, in general, than in the logistic mixed model, suggesting that the logistic mixed model may be slightly over-fitted. For example, according to the logistic mixed model, Mouse Drag tasks are 7.87 times as likely to be answered correctly, but according to the logistic lasso model, Mouse Drag tasks are only 6.02 times as likely to be answered correctly; similarly Right Click containing tasks in the mixed model are .04 times as likely to be answered correctly compared to .09 times as likely in the logistic lasso. These results suggest that while the logistic mixed model might be more appropriate for assessment purposes, as it additionally estimates task difficulty and subject ability, the logistic lasso model might be more appropriate for predicting the effects of component skills on success rates for new tasks.

6. GENERAL DISCUSSION

Together, our results suggest that not only are there specific Northstar tasks that are informative with regard to building an adaptive computer-based intervention for adults with low literacy skills but also that these tasks can themselves be decomposed into component skills that can be further used for this purpose. The main effects of Analysis 3 and coefficient rankings of Analysis 4 are consistent and complimentary with the proportion correct results in Analysis 1. The marginal main effect for Hardware Identify explains the high proportion correctness for identification tasks for mouse, keyboard, and headphone jack, and the main effect for Mouse Drag explains the high proportion correctness for recycling a file (dragging to the Recycle Bin), dragging, and scrolling (by dragging a scroll bar). These correctness-enhancing main effects are also reflected in odds ratios greater than one in Analysis 4. Similarly the main effects for Keyboard Function and Simple Typing explain the low proportion correctness for identifying various keys, typing web addresses, signing into email, and composing email, and these main effects are likewise reflected in odds ratios less than one in Analysis 4. In these cases we infer that the problem is not specific to the interface in question, e.g. email, but rather that there is a deficiency in a component skill needed for the task taking place in the context of that interface.

The implications for building adaptive computer-based interventions for adults with low literacy skills are clear. First, it is important to keep typing to a minimum, either by having users select response options or by using speech recognition. Second, right clicking should be eliminated or at least made optional. Third, icons should be close to icon archetypes. And finally, mouse dragging is a good skill around which to build user interaction. Interestingly, all of these implications seem to point to tablet and smartphone platforms, which have a minimum of typing (and built in speech interfaces), no right clicking, minimal icons in-app, and plenty of swiping/dragging. Moreover, smartphone ownership has been rapidly increasing – now 64% of households earning below $30 thousand own a smartphone [17]. It may be the case that deploying interventions on smartphones and tablets better makes use of both the computer literacy strengths and the material resources of low literacy adults.

7. ACKNOWLEDGMENTS

This research was supported by the Institute of Education Sciences (IES; R305C120001). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the author and do not represent the views of the IES.

8. REFERENCES

[1] M. Anderson and A. Perrin. 13% of Americans don't use the internet. Who are they? Technical report, Pew Research Center, 2016.

[2] D. Bakhtiari, A. Olney, and D. Greeberg. Computer literacy skills of adult learners. In preparation.

[3] J. A. Ballantine, P. M. Larres, and P. Oyelere. Computer usage and the validity of self-assessed computer competence among first-year business students. *Computers & Education*, 49(4):976 – 990, 2007.

[4] T. Barnes, D. Bitzer, and M. Vouk. Experimental analysis of the q-matrix method in knowledge discovery. In *International Symposium on Methodologies for Intelligent Systems*, pages 603–611. Springer, 2005.

[5] H. Beder and P. Medina. Classroom dynamics in adult literacy education. ncsall research brief. Technical report, National Center for the Study of Adult Learning and Literacy, 2002.

[6] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[7] A. C. Graesser, Z. Cai, W. O. Baer, A. M. Olney,

X. Hu, M. Reed, and D. Greenberg. Reading comprehension lessons in AutoTutor for the Center for the Study of Adult Literacy. In S. A. Crossley and D. S. McNamara, editors, *Adaptive Educational Technologies for Literacy Instruction.*, pages 288–293. Routledge, 2016. DOI: 10.4324/9781315647500 DOI: 10.4324/9781315647500.

[8] F. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Graduate Texts in Mathematics. Springer, 2001.

[9] B. E. John. Cogtool: Predictive human performance modeling by demonstration. In *Proceedings of the 19th Conference on Behaviour Representation in Modeling and Simulation*, pages 83–84, 2010.

[10] B. E. John and D. E. Kieras. The goms family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(4):320–351, 1996.

[11] B. E. John, K. Prevas, D. D. Salvucci, and K. Koedinger. Predictive human performance modeling made easy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 455–462, New York, NY, USA, 2004. ACM.

[12] S. Nakagawa and H. Schielzeth. A general and simple method for obtaining r2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2):133–142, 2013.

[13] N. D. L. Project, 2016.

[14] ProLiteracy. U.S. adult literacy facts. Technical report, 2017.

[15] F. Rijmen, P. D. Boeck, and K. U. Leuven. The random weights linear logistic test model. *Applied Psychological Measurement*, 26(3):271–285, 2002.

[16] A. A. Rupp and J. L. Templin. Unique characteristics of diagnostic classification models: A comprehensive review of the current state-of-the-art. *Measurement: Interdisciplinary Research and Perspectives*, 6(4):219–262, 2008.

[17] A. Smith. Record shares of americans now own smartphones, have home broadband. Technical report, Pew Research Center, 2017.

[18] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

Towards reliable and valid measurement of individualized student parameters

Ran Liu
Human-Computer Interaction Institute
Carnegie Mellon University
ranliu@cmu.edu

Kenneth R. Koedinger
Human-Computer Interaction Institute
Carnegie Mellon University
koedinger@cmu.edu

ABSTRACT

Research in Educational Data Mining could benefit from greater efforts to ensure that models yield reliable, valid, and interpretable parameter estimates. These efforts have especially been lacking for individualized student-parameter models. We collected two datasets from a sizable student population with excellent "depth" – that is, many observations for each skill for each student. We fit two models, the Individualized-slope Additive Factors Model (iAFM) and Individualized Bayesian Knowledge Tracing (iBKT), both of which individualize for student ability and student learning rate. Estimates of student ability were reliable and valid: they were consistent across both models and across both datasets, and they significantly predicted out-of-tutor pretest data. In one of the datasets, estimates of student learning *rate* were reliable and valid: consistent across models and significantly predictive of pretest-posttest gains. This is the first demonstration that statistical models of data resulting from students' use of learning technology can produce reliable and valid estimates of individual student learning rates. Further, we sought to interpret and understand what differentiates a student with a high estimated learning rate from a student with a low one. We found that learning rate is significantly related to estimates of student ability (prior knowledge) and self-reported measures of diligence. Finally, we suggest a variety of possible applications of models with reliable estimates of individualized student parameters, including a more novel, straightforward way of identifying wheel spinning.

Keywords

Explanatory models, model interpretability, individualized parameters, 3, Additive Factors Model, individualized Bayesian Knowledge Tracing

1. INTRODUCTION

In Educational Data Mining, statistical models are typically evaluated based on fit to overall data and/or predictive accuracy on test data. While this is an important initial step in evaluating the contributions of advancements in statistical and cognitive modeling, research in the field could benefit from greater efforts to ensure that models are reliable and valid. More reliable and valid models offer more explanatory power, contributing to the advancement of learning science. They also inspire greater confidence that deploying model advancements in future tutoring systems will genuine result in the hypothesized improvements to learning.

Some recent work has been done towards interpreting, validating, and acting upon cognitive/skill modeling improvements [7, 8, 10, 11, 17]. Educational data mining efforts oriented around personalizing student constructs [3, 12, 13, 14, 18], however, have remained focused on improving predictive accuracy and/or demonstrating hypothetical time savings. Little has been done to validate or understand the estimates that models with individualized or clustered student parameters produce. Anecdotally, efforts to do so have shown that these individualized student parameter estimates, or discovered student clusters, are often difficult to interpret.

It is especially critical to examine the reliability and validity of parameter estimates for modeling advancements that dramatically increase the parameter count, as is generally true for individualized student-parameter models. More parameters create greater degrees of freedom and increase the likelihood that the model may be underdetermined by the data.

We focus on the question: To what degree can we trust a model's parameter estimates to correctly represent the constructs they are supposed to?

Key to expecting reliable, valid estimates of student-level constructs is not just big data in the "long" sense, but big data in the "deep" sense. Oftentimes, the datasets used in secondary analyses in EDM are large in terms of total number of students (or total observations) but highly sparse in terms of observations per skill, per student. These features make it difficult to get reliable measurements of constructs at the individual student level, particularly constructs related to learning over time.

Here, we collected two datasets from a sizable student population (196 students) with excellent "depth" – that is, many observations for each skill for each student. We then fit two models that individualize for student ability and student learning rate (the Individualized-slope Additive Factors Model [9] and Individualized Bayesian Knowledge Tracing [18]). We assess the models' fit to data and predictive accuracy. We also move beyond these metrics to examine the reliability of the models' estimates of student ability and student learning rate. Additionally, we externally validate the parameter estimates against out-of-tutor assessment data.

We further interpret and understand the constructs by visualizing representative student learning trajectories, examining the relationship between estimated student ability and student learning rate, and the relationship between those constructs and self-reported data on motivational attributes. Finally, we propose some useful applications of reliable and valid individualized student-parameter models, including a new way to detect wheel spinning.

2. PRIOR WORK

Prior work on individualizing student parameters has focused on variants of Bayesian Knowledge Tracing (BKT) [3]. This work includes modeling the parameters separately for each individual student instead of separately for each skill [3], individualizing the P(Init) ("initial knowledge") parameter for each student [13], and individualizing both P(Init) and P(Learn) ("learning rate") to the

base BKT model [18]. These models have generally focused on assessing predictive accuracy improvements relative to their respective non-individualized baseline models.

There have also been some "time savings" analyses [12, 18] that evaluate the hypothetical real world impact that individualizing statistical model fits could have. These analyses report the effect of fitting individualized BKT models, compared to traditional BKT, on the hypothetical number of under- and over- practice attempts that would be predicted for each student. Results generally have indicated that many more practice opportunities are needed for models to infer the same level of knowledge when using whole-population parameters rather than individual student parameters. These analyses show that individualized models differ in their hypothetical decision points if they were to be applied to drive mastery-based learning, but they do not in and of themselves interpret the individualized parameter estimates, nor do they assess the reliability and validity of such estimates.

In a previous effort to better understand individualized student learning rate parameters [9], we examined predictive accuracy and parameter reliability in an extension of the Additive Factors Model [2] applied to existing educational datasets. We did not find evidence that individualizing student rate parameters consistently improved predictive accuracy improvements, nor could we validate the parameter estimates on out-of-tutor assessment data. However, the datasets we analyzed either contained a small number of students or were largely sparse in observations for student-skill pairs, with the exception of two datasets. These two datasets happened to be the ones on which the Individualized-slope Additive Factors Model *did* achieve higher predictive accuracy. Thus, we wondered if the sparsity of the datasets were the primary limitation, rather than the modeling advancement itself. This idea is corroborated by the fact that pooling students into "groups" rather than generating individualized estimates worked well on those datasets [9].

For the present modeling work, we collected our own data in order to ensure the data features that we believe are necessary for reliable, valid, and potentially meaningful estimates of constructs at the individual student level.

3. METHODS

It is common in EDM to do secondary analyses across multiple datasets. However, it can be difficult to find datasets that (1) contain a sizable number of students, (2) contain many observations for each skill for each student (i.e., are not sparse), (3) contain students spanning a range of abilities in the domain covered by the tutor, and (4) contain data from out-of-tutor assessment data that is well-mapped to the content in the tutor.

For the present work, we wanted to use as close to an "ideal" dataset as possible for estimating student parameters. We collected our own dataset with a sizable number of students (196), many observations (5-50, depending on the skill) for each skill for each student. In addition, we ensured that a wide range of student ability levels was represented in our data to allow for the possibility that models could capture this variability.

3.1 Data Collection

196 students, spanning 10 classes taught by three different teachers, enrolled in high school geometry participated in two studies conducted about a month apart. A range of student abilities were included in the study. Two of the 10 classes were "Honors" and three of the 10 classes were "Inclusion". Honors classrooms are intended for students who have strong theoretical interests and abilities in mathematics. Inclusion classrooms are "general education" classrooms designed to provide the opportunity for individuals with disabilities and special needs to learn alongside their non-disabled peers.

Students spent five consecutive days participating in each study during their regular geometry class periods. On the first and last days, they took a computerized pretest and posttest, respectively. During the middle three days, they worked within an intelligent tutoring system [19] designed to give them practice on their current chapter's content. This procedure applied to both studies, one of which covered the students' Chapter 3 content (Parallel Lines Cut by a Transversal, Angles & Parallel Lines, Finding Slopes of Lines, Slope-Intercept Form, Point-Slope Form) and the other of which covered the students' Chapter 4 content (Classifying Triangles, Finding Measures of Triangle Sides & Angles, Triangle Congruence Properties). Figure 1 shows an example problem interface from the intelligent tutoring system, which was designed using Cognitive Tutor Authoring Tools [1].

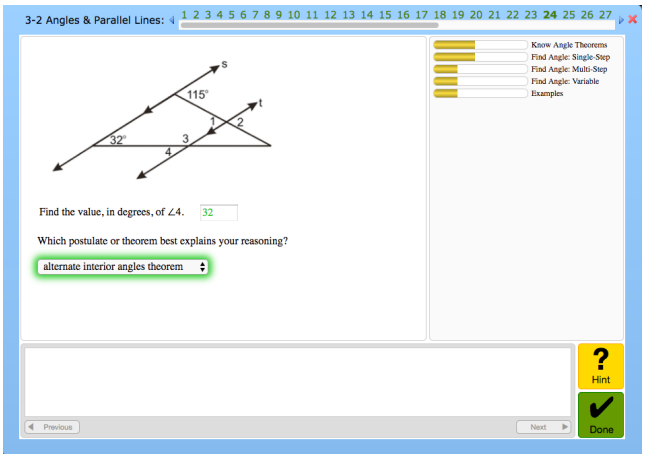


Figure 1. Example problem interface from the intelligent tutoring system used for data collection.

We also collected self-report survey data on motivational factors falling along three dimensions. These were Competitiveness (e.g., "In this unit, I am striving to do well compared to other students" and "In this unit, I am striving to avoid performing worse than others"), Effort (e.g., "I am striving to understand the content of this unit as thoroughly as possible" and "I work hard to do well in this class even if I don't like what we are doing"), and Diligence (e.g., "when class work is difficult, I give up or only study the easy parts" [inverted scale] and "I am diligent"). Self-report measures were indicated on a Likert scale from 1-7.

A key reason we collected two datasets, covering two distinct chapters of the curriculum, is that we were interested in investigating the consistency of student-level parameter estimates across different content, time, and contexts. We discuss this further, along with preliminary results, in Section 4.4.1.

3.2 Statistical Models

3.2.1 The Individualized-slope Additive Factors Model (iAFM)

The Additive Factors Model (AFM) [2] is a logistic regression model that extends item response theory by incorporating a growth or learning term.

$$\ln\left(\frac{p_{ij}}{1-p_{ij}}\right) = \theta_i + \sum_{k \in KCs} Q_{jk}(\beta_k + \gamma_k T_{ik}) \quad (1)$$

This statistical model (Equation 1) gives the probability p_{ij} that a student i will get a problem step j correct based on the student's baseline ability (θ_i), the baseline easiness (β_k) of the required knowledge components on that problem step (Q_{jk}), and the improvement (γ_k) in each required knowledge component (KC) with each additional practice opportunity. This KC slope, or "learning rate," parameter is multiplied by the number of practice opportunities (T_{ik}) the student already had on it. Knowledge components (KCs) are the underlying facts, skills, and concepts required to solve problems [6].

Individualized-slope AFM (iAFM) builds upon this baseline model by adding a per-student learning rate parameter (δ_i). This parameter represents the improvement (δ_i) by student i with every additional practice opportunity with the KCs required on problem step j.

$$\ln\left(\frac{p_{ij}}{1-p_{ij}}\right) = \theta_i + \sum_{k \in KCs} Q_{jk}(\beta_k + \gamma_k T_{ik} + \delta_i T_{ik}) \quad (2)$$

The KC and student learning rate parameters are both multiplied by the number of opportunities (T_{ik}) the student already had to practice that KC.

3.2.2 Individualized Bayesian Knowledge Tracing (iBKT)

Bayesian Knowledge Tracing (BKT [3]) is an algorithm that models student knowledge as a latent variable using a Hidden Markov Model. The goal of BKT is to infer, for each skill, whether a student has mastered it or not based on his/her sequence of performance on items requiring that skill. It assumes a two-state learning model whereby each skill is either *known* or *unknown*. There are four parameters that are estimated in a BKT model: the initial probability of knowing a skill a priori – p(Init), the probability of a skill transitioning from not known to known state after an opportunity to practice it – p(Learn), the probability of slipping when applying a known skill – p(Slip), and the probability of correctly guessing without knowing the required skill – p(Guess). Fitting BKT produces estimates for each of these four parameters for every skill in a given dataset. BKT models are usually fit using the expectation maximization method (EM), Conjugate Gradient Search, or discretized brute-force search.

Individualized Bayesian Knowledge Tracing (iBKT [18]) builds upon this baseline BKT model by individualizing the estimate of the probability of initially knowing a skill, p(Init), and the transition probability, p(Learn), for each student. To accomplish the student-level individualization of these parameters, each of them is split into skill- and student-based components that are summed and passed through a logistic transform to yield the final parameter estimate. Details on the decomposition of p(Init) and p(Learn) into skill- and student-based components are described in [18].

4. RESULTS

4.1 Model Fit & Predictive Accuracy

As a first pass evaluation of the two individualized models, we assessed them using Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), which are standard metrics for model comparison, and 10 independent runs of split-halves cross validation (CV). Although 10-fold cross validation has been popular in the field, [4] showed that it has a high type-I error due to high overlap among training sets and recommended at least 5 replications of 2-fold CV instead.

Here, the comparison of interest is each individualized model against its non-individualized counterpart. We do not encourage a literal comparison between the predictive accuracies of the two classes of models due to differences in whether they use incoming test data towards their predictions on later test data (BKT/iBKT do, and AFM/iAFM do not).

Both iAFM and iBKT outperform their non-individualized counterparts by all metrics, with the exception of BKT having a better BIC value than iBKT for the Chapter 4 dataset. This is not surprising, as BIC is known to over-penalize for added parameters. We recommend cross validation as a better indicator that iBKT is the true better fitting model in this case.

Counter to the majority of findings reported in [9], iAFM achieved higher predictive accuracy than AFM in both datasets here. This further supports the idea that the "depth" of the dataset is a critical factor in whether an individualized student-parameter model can explain unique variance in the data.

Table 1. Summary of Model Fit and Predictive Accuracy metrics comparing AFM vs. iAFM and BKT vs. iBKT. Cross-validation values are mean RMSE values across 10 runs, with standard deviations included in parentheses.

Data Set	Model	AIC	BIC	CV *Test* RMSE (10-Run Average)
Ch. 3	AFM	57229	57283	0.38440 (0.0039)
	iAFM	**55931**	**56003**	**0.37868 (0.0044)**
	BKT	66714	67473	0.4222 (0.0005)
	iBKT	**56325**	**60479**	**0.3777 (0.0006)**
Ch. 4	AFM	18059	18106	0.41037 (0.0048)
	iAFM	**17863**	**17925**	**0.40789 (0.0050)**
	BKT	19908	**20376**	0.44091 (0.0014)
	iBKT	**18285**	21809	**0.40725 (0.0018)**

4.2 Reliability of Student Parameters

Next, we examined the degree to which we can rely on these parameters to reasonably estimate the constructs that they should be estimating. We believe that a strong relationship between the parameter estimates of two statistical models with entirely different architectures is a high bar for testing reliability. That is, if a student genuinely displayed evidence of high overall ability in a dataset (relative to his/her peers), then both iAFM and iBKT should estimate that to be the case.

Because of known and observed nonlinear relationships between logistic regression and Bayesian Knowledge Tracing parameter estimates, we measured correlation based on Spearman's coefficient (r_s), which is based on rank order.

We observed strong and statistically significant correlations between iAFM Student Intercept and iBKT Student p(Init) parameter estimates (Figure 2, top row). We also observed a strong and statistically significant correlation between iAFM Student Slope and iBKT Student p(Learn) parameter estimates for one of the two datasets (Chapter 4). This correlation was much milder, though still significant, for the other dataset (Chapter 3).

We hypothesize that this difference between datasets may be due to the presence of more difficult KCs in Chapter 4. A dataset with more difficult items should provide more sensitive measures of individual differences in improvement, since it avoids ceiling effects. Indeed, this was the case: the mean KC easiness parameter estimate (β_k) for chapter 4 was 0.799 (which translates to a

probability of 0.69), compared to 1.253 for chapter 3 (which translates to a probability of 0.78). When students are practicing many opportunities at ceiling (which was the case in particular for chapter 3, based on exploratory analyses of the data), the individualized models will often assign them a lower "learning rate" due to an essentially flat learning trajectory.

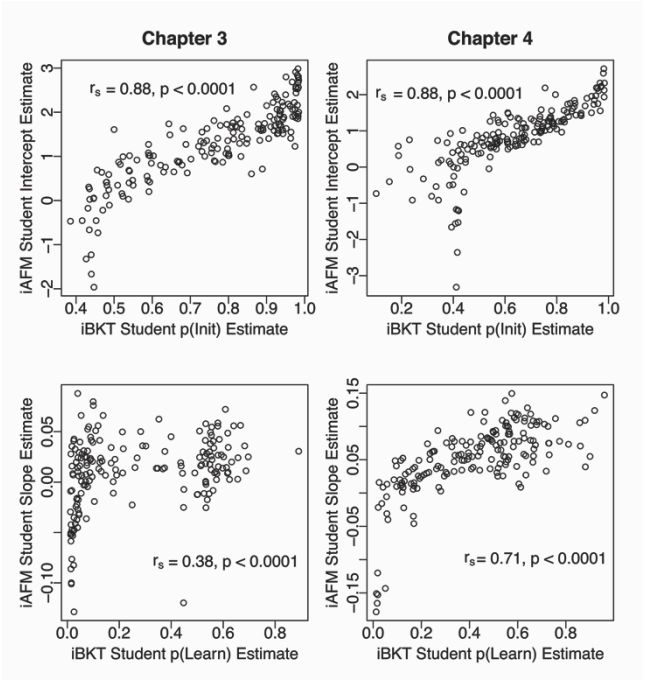


Figure 2. Relationships between iAFM Student Intercept and iBKT Student p(Init) parameter estimates (top row), and between iAFM Student Slope and iBKT Student p(Learn) parameter estimates (bottom row), for the two datasets.

4.3 Validity of Student Parameters

To assess the validity of student parameter estimates, we related them to out-of-tutor assessments of the relevant student constructs. In this case, we validated parameter estimates using pretest and posttest assessment data collected in the study.

4.3.1 Estimates of Student Ability

The Student Intercept (θ_i) parameter of iAFM and the Student p(Init) parameter of BKT are designed to estimate baseline student ability, as least for the knowledge domain represented in the dataset. To validate the models' estimates of this construct, we examined relationships between the model estimates and students' pretest scores, which are an out-of-tutor assessment of student initial ability for the skills covered by the tutor.

We report standard Pearson correlation coefficients here, since the relationships between pretest scores and the parameter estimates did not appear to be particularly nonlinear.

Figure 3 illustrates a summary of these relationships. Both models' estimates of the student ability construct were strongly and significantly correlated with pretest scores.

In addition, adding an individualized student slope *improved* the validity of the model's estimate of student ability (a parameter that's modeled in both AFM and iAFM). We compared the correlations between AFM's intercept estimates to pretest scores (Chapter 3: r = 0.62, p < 0.0001, Chapter 4: r = 0.58, p < 0.0001) to iAFM's intercept estimate / pretest score correlations (Chapter 3: 0.74, p < 0.0001, Chapter 4: r = 0.66, p < 0.0001).

This has several interesting implications for educational applications. First, it suggests that formative assessment via modeling of process data as learning unfolds is a reasonable method of assessment.

It also suggests that detailed assessment data (e.g., from a pretest) could be used to reasonable effect to improve different students' "on-line" estimates of students' knowledge of KCs. For example, combining KC parameter estimates (derived from model-fitting to prior domain-relevant data) with student intercept priors based on pretest assessment data would allow a model like AFM to generate individualized predictions of how much each student needs to practice to reach mastery.

In addition, these results suggest that individualized BKT models could use pretest assessment data to "set" reasonably valid student-specific p(Init) values before collecting any within-tutor data from those students.

In considering the degree to which these results may generalize, it is important to note that the pretests in the present datasets were specifically designed to map closely to the practice problems in the intelligent tutor. Pretests contained 1-2 questions for each KC that was practiced in the tutor, and the items were similar to those encountered within the tutor.

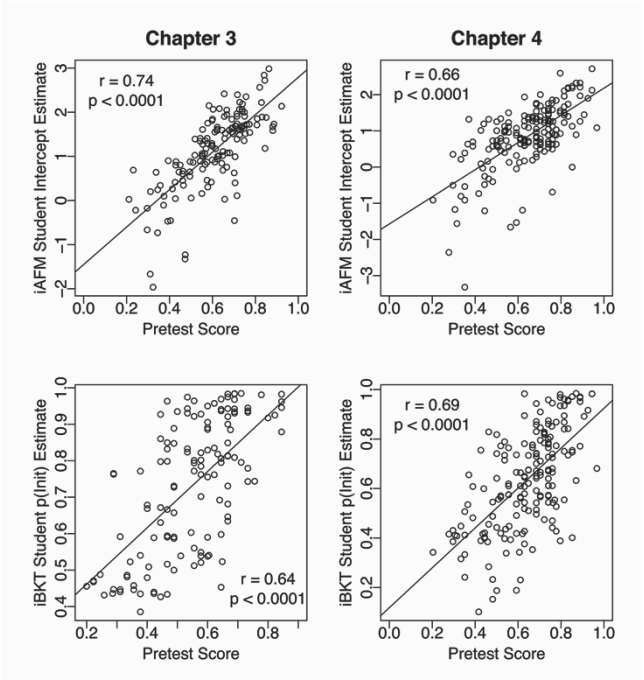


Figure 3. Relationships between out-of-tutor pretest scores and iAFM/iBKT estimates of student ability based on within-tutor data.

4.3.2 Estimates of Student Learning Rate

Given that the only external assessment data collected were a pretest and posttest, we sought to validate the construct of student learning rate (as estimated by the models) on pretest-posttest gains. Students were given roughly the same amount of time to engage with the tutors, so those with accelerated learning rates might be expected to gain more knowledge in the time available.

Thus, we examined the degree to which student learning rate estimates predicted pretest-posttest gains while controlling for pretest scores. We controlled for pretest scores because they have been shown to negatively predict learning gains due to assessment

ceiling effects. That is, students who start out performing well on the pretest have less "room for improvement".

For the Chapter 3 dataset, iAFM Student Slope (δ_i) estimates did not significantly predict learning gains. In a linear regression predicting pretest-posttest gains, pretest scores were a significant predictor (β=-0.189, p=0.005) and Student Slope estimates were not (β=0.396, p=0.144). iBKT Student p(Learn) estimates did not significant predict learning gains. In a linear regression predicting pretest-posttest gains, pretest scores were a significant predictor (β=-0.226, p=0.005) and Student Slope estimates were not (β=0.062, p=0.218).

For the Chapter 4 dataset, iAFM Student Slope (δ_i) estimates significantly predict learning gains. In a linear regression predicting pretest-posttest gains, pretest scores (β=-0.641, p<0.0001) and Student Slope estimates (β=0.576, p=0.007) were both significant predictors. iBKT Student p(Learn) estimates also significantly predict learning gains. In a linear regression predicting pretest-posttest gains, pretest scores (β=-0.645, p<0.0001) and p(Learn) estimates (β=0.133, p=0.004) were both significant predictors.

For one of the two units (Chapter 4), we observed that student learning rate estimates were validated on external assessments of learning gain. Interestingly, this is the same unit for which we observed a strong cross-model reliability in student learning rate estimates. Thus, we have converging evidence that student learning rates estimates for the Chapter 4 dataset are both reliable and valid.

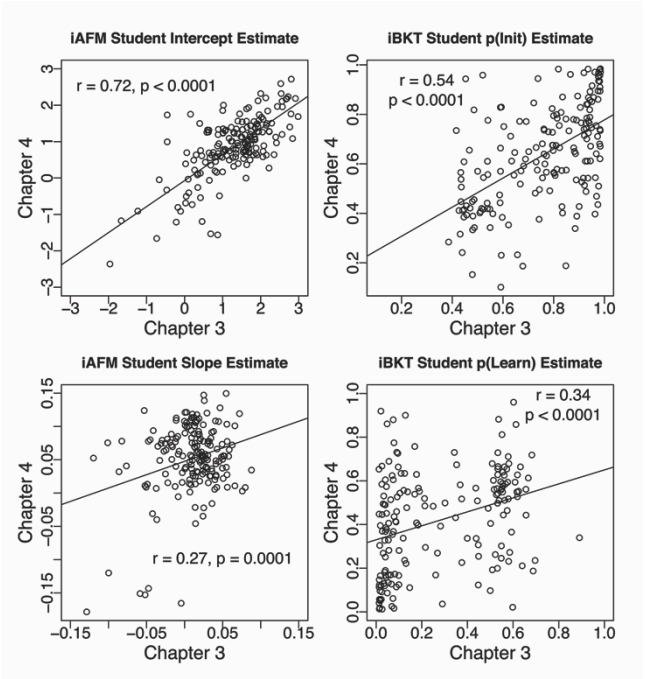


Figure 4. Relationships between student parameter estimates across the two datasets (same student population).

4.4 Towards Understanding & Using Student Parameter Estimates

4.4.1 Consistency of individual student constructs across datasets

A core motivating question for collecting two datasets on the same group of students was: How consistent are iAFM and iBKT model estimates of the student ability and student learning rate constructs across units?

Figure 4 summarizes this relationship. Estimates of student ability are fairly consistent, especially as estimated by iAFM. It seems sensible to interpret this as suggesting that overall student ability on Chapter 3 content is strongly related to overall student ability on Chapter 4 content, as we have shown estimates of student ability to be both reliable and valid.

Estimates of student learning rate are less consistent. This may either be due to the fact that Chapter 3 estimates of student learning rate were neither very reliable nor very valid. Alternatively, the differences in student learning rate estimates across the two chapters may also be due to the fact that students genuinely learn different material at different rates. Unfortunately, we cannot resolve this question with the present data. We are currently collecting more datasets from this same group of students. If we obtain more reliable and valid student learning rate estimates in future data from this group of students, we can more confidently address this question in future research.

4.4.2 Understanding student learning rate estimates

Given that we established the reliability and validity of iAFM and iBKT's parameter estimates for the Chapter 4 dataset were reasonably reliable and valid, we sought to dig deeper into the explanatory power of these estimates. To this end, we conducted exploratory analyses on the Chapter 4 data to (1) visualize the learning trajectories of students with the highest vs. lowest estimated learning rates, (2) understand the relationships between estimated learning rates and prior-knowledge and motivational factors, and (3) understand the degree of variability in estimated learning rate across students.

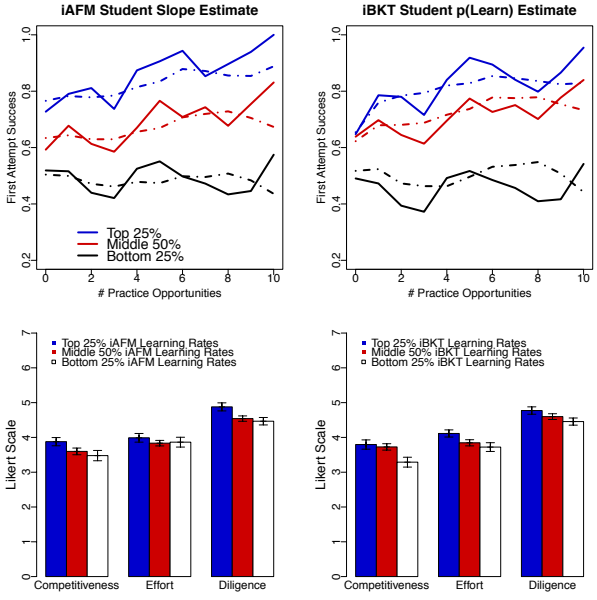


Figure 5. *Top Row*: Early-opportunity learning trajectories of students, grouped based on iAFM (Left) and iBKT (Right) estimated learning rates. Solid lines are actual data; dotted lines are each respective model's predicted performance. *Bottom Row*: Mean self-report Likert scale ratings of questions measuring dimensions of competitiveness, effort, and diligence. Grouped based on iAFM (Left) or iBKT (Right) estimated learning rates. Error bars show standard errors on the means.

Figure 5 (top row) shows the aggregate learning trajectories for students split based either on their iAFM Student Slope estimates (top left) or their iBKT Student p(Learn) estimates (top right). The top 25% of student parameter estimates are plotted in blue, the middle 50% (between 1st and 3rd quartiles) are plotted in red, and the lower 25% are plotted in black. Dotted lines represent each respective model's *predicted* earning trajectories.

One striking pattern, especially in the iAFM learning trajectories (top left), is the apparent relationship between average success on initial practice opportunities (i.e., prior knowledge) and estimated learning rate through the remaining opportunities. This observation is corroborated by a strong and significant correlation between iAFM Student Intercepts and iAFM Student Slopes ($r=0.78$, $p<0.0001$). One might interpret this to suggest that students who enter into the tutor with greater prior knowledge will be poised to gain more from the tutor (i.e., "the rich get richer"). Alternatively, students may have higher overall knowledge *because* they are fast learners. There may also be individual trait-based variables that positively drive both learning rate and overall achievement.

To explore the relationships between measures of traits relevant to learning, we analyzed self-report survey data grouped by three factors (as described in Section 3.1): Competitiveness, Effort, and Diligence. The relationship between these measures and the high, medium, and low learning rate estimates from iAFM and iBKT are shown in Figure 5 (bottom row). There appears to be a relationship between the means of each self-report measure and the general range that the learning rate estimate falls in.

We analyzed the continuous relationship between students' mean self-report rating along each dimension and their iAFM learning rate estimates. In a linear regression predicting iAFM Student Slopes, Competitiveness and Effort were not significant predictors but Diligence ($\beta=0.016$, $p=0.007$) was. In a similar linear regression predicting iAFM Student Intercepts, again Diligence was the only significant predictor ($\beta=0.02$, $p=0.04$). Thus, among self-reported measures, the strongest dimension predicting both student ability/prior knowledge *and* student learning rate was the Diligence measure. Future work using causal modeling is warranted to discover the true nature of causality among these student-level constructs.

Finally, we investigated the degree of variability in estimated learning rate across students. The first quantile of student learning rates from iAFM is 0.03 logits and the third quantile of rates from iAFM is 0.08 logits. These can be conceptualized as canonical "slow" and "fast" learners. If we were to assume starting at around 70% performance (which comes from the model's global intercept estimate), it would take the "slow" (0.03 logits) student approximately 25 opportunities to reach mastery (defined as 85%, the performance equivalent of a p(Know)=0.95, factoring in the guess and slip probabilities we used in the actual tutor). It would take the "fast" (0.08 logits) student approximately 11 opportunities to reach the same place.

4.4.3 Identifying wheel spinners

The current definition of "wheel spinning" put forth in the Educational Data Mining community is the "phenomenon in which a student has spent a considerable amount of time practicing a skill, yet displays little or no progress towards mastery" [5]. There has been some controversy around the ideal way to measure mastery (e.g., 3 corrects in a row vs. reaching a certain p(Know) in knowledge tracing). Furthermore, some students may be classified as wheel spinners based on not mastering in a certain number of opportunities but they may still be making progress.

We propose that reliable and validated estimates of individual student learning rate parameters, combined with KC learning rate parameters, could be used to estimate wheel spinning student/KC pairs in way that is agnostic to mastery status. Specifically, if the combined student and KC learning rate parameters in iAFM predict *no* improvement or negative improvement across additional practice opportunities, and aren't already at a high level of performance on their first opportunity (here we considered this to be 80% or above), we could consider the student to be wheel spinning on the KC. This method of estimating wheel spinning would be particularly useful for datasets with sparse data on some student-KC pairs, as it is not performance-dependent after the model has been fit to the full dataset.

Based on this operationalized definition, we found that approximately 15% of student-KC pairs in the Chapter 4 dataset are estimated to be wheel spinning. That is, those students are not making progress on those KCs. This is a substantially lower estimate than the 25% reported by a recent wheel spinning detector in [5]. An interesting route for future work would be to do a direct comparison of the wheel spinning detector presented in [5] and our proposed student/KC learning rate identifier within the same dataset. This would allow for testing the possibility that some students who are still making progress, albeit extremely slowly, may be prematurely labeled as "wheel spinners" by [5].

5. SUMMARY & LIMITATIONS

Previous efforts towards more explanatory, interpretable, and actionable modeling advancements in the realm of skill/knowledge component model discovery have been promising in their potential and demonstrated impact on learning science and education. The present paper represents a novel effort to bring these deeper modeling approaches, focused on ensuring explanatory power, to the realm of individualized student-parameter models.

Towards improving the reliability and validity of individualized student estimates, we collected two datasets from the same student population. Both datasets were "deep" along the dimension of student-KC observations. We fit iAFM and iBKT to both datasets and showed that the models outranked their non-individualized counterparts in terms of fit to data and predictive accuracy. Importantly, we moved beyond these metrics to show that estimates of student ability were highly reliable (iAFM and iBKT yielded strongly correlated estimates) and valid (estimates significantly predicted pretest data).

This demonstration of confidence in the student ability estimates from iBKT, but even more so iAFM, has promising implications for the possibility of individualizing the student models that determine mastery in intelligent tutoring systems at *least* in terms of overall student ability/knowledge. Our results also suggest that it would be reasonable to fix such student ability parameters, or set priors on them, based on either well-mapped pretest assessment data or prior (deep) data from those students' learning.

We also showed that estimates of student learning rate per practice opportunity were reliable and valid in one of the two datasets (Chapter 4). This is the first evidence, to our knowledge, of obtaining both reliable and valid student learning rates through a statistical model with *individualized* student parameters. We believe that this success is largely related to the amount and quality of per-student data we collected.

With the confidence of having reliable and valid parameter estimates, we then proceeded to further investigate potential explanations for differences in student learning rates within the

Chapter 4 dataset. We found a strong and significant relationship between student ability and improvement rate as well as an additional effect of diligence, based on self-report measures. Further research is warranted to distill the causal relationships between these constructs.

Knowing that a model's estimates of individualized student parameters not only fit data well, but are reliable and valid, provides greater confidence for applying the model to (1) interpret the parameter estimates to understand characteristics of students, and (2) use the model to individualize the trajectory of mastery estimation for future students.

Even though both iBKT and iAFM outperformed their non-individualized counterparts in predicting performance in the Chapter 3 dataset, we did not find strong evidence of reliability and validity of the student-specific parameter estimates. Thus, we did not rely on that dataset to help us understand individual differences in learning rates. For the same reason, we could not confidently attribute the differences, in estimated student learning rates across the datasets, to *true* differences in students' learning rates for the two chapters' material.

Although considering reliability and validity of models' parameter estimates sets a higher bar than predictive accuracy for evaluating modeling advances, we believe those to be important characteristics of a model that is to be explanatory, interpretable, and/or actionable. Here, we have demonstrated that with a sufficiently good dataset, iAFM and iBKT are individualized student models that *can* produce reliable and valid parameter estimates.

Since our present work was limited to two datasets on one population of students, it is unclear the degree to which our modeling results will generalize, especially given that at least iAFM does not produce reliable, valid parameter estimates on more sparse datasets [9]. In addition, these results are limited to two specific statistical models produce individualized estimates student-level parameters, with a particular focus on individual differences in learning rate. There are other classes of models that could be extended to estimate differences in learning rate: for example, producing individualized estimates of the differential effects of success versus failure [15]. This would be an interesting focus for future work on this topic.

Nevertheless, we have laid a foundation of methodology by which reliability and validity of parameter estimates, whether student- or KC-level, can be assessed. We have also demonstrated ways of using the reliable and valid student parameter estimates from iAFM and iBKT to yield interesting insights about student learning.

6. ACKNOWLEDGMENTS

We thank the Institute of Education Sciences for support to RL (training grant #R305B110003) and the National Science Foundation for support to Carnegie Mellon University's LearnLab (#SBE-0836012).

7. REFERENCES

[1] Aleven, V., Sewall, J., McLaren, B.M., and Koedinger, K.R. (2006). Rapid authoring of intelligent tutors for real-world and experimental use. In *Proceedings of the 6th ICALT*. IEEE, Los Alamitos, CA, pp. 847-851.

[2] Cen, H., Koedinger, K.R., & Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. Intelligent Tutoring Systems, 164-175.

[3] Corbett, A.T., & Anderson, J.R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction, 4, 253-278.

[4] Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation, 10(7), 1895–1923.

[5] Gong, Y. & Beck, J. (2015). Towards Detecting Wheel-Spinning: Future Failure in Mastery Learning. In Proceedings of Learning At Scale '15.

[6] Koedinger, K.R., Corbett, A.C., & Perfetti, C. (2012). The Knowledge-Learning-Instruction (KLI) framework: Bridging the science-practice chasm to enhance robust student learning. Cognitive Science, 36(5), 757-798.

[7] Koedinger, K.R., McLaughlin, E.A., & Stamper, J.C. (2012). Automated Student Model Improvement. 5th International Conference on EDM.

[8] Koedinger, K. R., Stamper, J. C., McLaughlin, E. A., & Nixon, T. (2013). Using data-driven discovery of better cognitive models to improve student learning. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED '13), 9–13 July 2013, Memphis, TN, USA (pp. 421–430). Springer.

[9] Liu, R., & Koedinger, K. R. (2015). Variations in learning rate: Student classification based on systematic residual error patterns across practice opportunities. In O. C. Santos, J. G. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. M. Luna, C. Mihaescu, P. Moreno, A. Hershkovitz, S. Ventura, & M. Desmarais (Eds.), Proceedings of the 8th International Conference on Education Data Mining (EDM2015), 26–29 June 2015, Madrid, Spain (pp. 420–423). International Educational Data Mining Society.

[10] Liu, R., & Koedinger, K. R. (under review). Closing the loop: Automated data-driven skill model discoveries lead to improved instruction and learning gains.

[11] Liu, R., Koedinger, K. R., & McLaughlin, E. A. (2014). Interpreting model discovery and testing generalization to a new dataset. In J. Stamper, Z. Pardos, M. Mavrikis, & B. M. McLaren (Eds.), Proceedings of the 7th International Conference on Educational Data Mining (EDM2014), 4–7 July, London, UK (pp. 107–113). International Educational Data Mining Society.

[12] Lee, J.I., & Brunskill, E. (2012). The Impact on Individualizing Student Models on Necessary Practice Opportunities. 5th International Conference on EDM.

[13] Pardos, Z.A., & Heffernan, N.T. (2010). Modeling individualization in a bayesian networks implementation of knowledge tracing. User Modeling, Adaptation, and Personalization, 255-266.

[14] Pardos, Z. A., Trivedi, S., Heffernan, N. T., & Sárközy, G. N. (2012). Clustered knowledge tracing. In S. A. Cerri, W. J. Clancey, G. Papadourakis, K.-K. Panourgia (Eds.), Proceedings of the 11th International Conference on Intelligent Tutoring Systems (ITS 2012), 14–18 June 2012, Chania, Greece (pp. 405–410). Springer.

[15] Pavlik, P.I., Cen, H., & Koedinger, K.R. (2009). Performance factors analysis–a new alternative to knowledge tracing. AIED, 531–538.

[16] Shmueli, G. (2010). To explain or to predict? *Statistical Science*, *25*(3), 289–310. doi:10.1214/10-STS330

[17] Stamper, J., & Koedinger, K. R. (2011). Human-machine student model discovery and improvement using data. *Proceedings of the 15th International Conference on*

Artificial Intelligence in Education (AIED '11), 28 June–2 July, Auckland, New Zealand (pp. 353–360). Springer.

[18] Yudelson, M.V., Koedinger, K.R., & Gordon, G.J. (2013). Individualized bayesian knowledge tracing models. AIED, 171-180.

[19] VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*, 227–265.

The Misidentified Identifiability Problem of Bayesian Knowledge Tracing

Shayan Doroudi
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15206
shayand@cs.cmu.edu

Emma Brunskill
Computer Science Department
Stanford University
Stanford, CA 94305
ebrun@cs.stanford.edu

ABSTRACT

In this paper, we investigate two purported problems with Bayesian Knowledge Tracing (BKT), a popular statistical model of student learning: *identifiability* and *semantic model degeneracy*. In 2007, Beck and Chang stated that BKT is susceptible to an *identifiability problem*—various models with different parameters can give rise to the same predictions about student performance. We show that the problem they pointed out was not an identifiability problem, and using an existing result from the identifiability of hidden Markov models, we show that under mild conditions on the parameters, BKT is actually identifiable. In the second part of the paper, we discuss a problem that has been conflated with identifiability, but which actually does arise when fitting BKT models, *semantic model degeneracy*—the model parameters that best fit the data are inconsistent with the conceptual assumptions underlying BKT. We give some intuition for why semantic model degeneracy may arise by showing that BKT models fit to data generated from alternative models of student learning can have semantically degenerate parameters. Finally, we discuss the potential implications of these insights.

Keywords

Bayesian Knowledge Tracing, identifiability, semantic model degeneracy

1. INTRODUCTION

Bayesian Knowledge Tracing (BKT) is a popular model of student learning that tries to predict the probability that a student knows a skill and the probability that a student will answer questions based on the skill correctly. The BKT model is a two state hidden Markov model (HMM) that posits students have either mastered a skill or not, and at every practice opportunity, a student who has not mastered the skill has some chance of attaining mastery. If a student has mastered a skill, they will answer a question correctly unless they "slip" with some (ideally small) probability, and if the student has not mastered the skill, they can only guess correctly with some (ideally small) probability. In 2007, Beck and Chang stated that BKT is not identifiable, meaning that different settings of the four BKT parameters can lead to identical predictions about a student's performance [7]. Whether or not BKT is identifiable is an important issue, because if BKT is not identifiable, it means that we would fundamentally need other criteria (beyond accurately modeling student performance data) to fit BKT models.

However, in this paper, we show that BKT is actually an identifiable model, under mild conditions on the parameters that should always be satisfied in practical settings. This result follows from BKT being a special case of a hidden Markov model and therefore it inherits identifiability results that prior work has proven for HMMs. This implies no additional criteria beyond predictive accuracy are needed to identify a single BKT model that best explains observed student performance, under the assumption that learning can accurately be modeled by a BKT. We then describe three potential issues with BKT models that may have been misconstrued as an identifiability problem in the literature. Note that our goal is by no means to criticize prior researchers, as such researchers helped identify some important limitations of Bayesian Knowledge Tracing, but these limitations do not stem from a lack of identifiablity.

In the second part of this paper, we focus on one of the issues that has been conflated with identifiability, but which actually does arise when fitting BKT models, *semantic model degeneracy*—the model parameters that best fit the data are inconsistent with the conceptual assumptions underlying BKT. We give a critical look at the types of semantic model degeneracy in the literature and then give some intuition for why this problem may arise by showing that BKT models fit to data generated from alternative models of student learning can have degenerate parameters. We further show that fitting models to sequences of different lengths generated from the same underlying model can result in different forms of semantic degeneracy. We show that these insights can have important implications on how these models should be used.

2. BAYESIAN KNOWLEDGE TRACING

The Bayesian Knowledge Tracing model is a two-state hidden Markov model that keeps track of the probability that a student has mastered a particular skill and the probability that the student will be able to answer a question on that skill correctly over time. At each practice opportunity $i \geq 1$ (i.e., when a student has to an answer a question corresponding to the skill), the student has a latent knowledge state $K_i \in \{0, 1\}$. If the knowledge state is 0, the student has not mastered the skill, and if it is 1, then the student has mastered it. The student's answer can either be correct or incorrect: $C_i \in \{0, 1\}$ (where 0 corresponds to incorrect and 1 corresponds to correct). After each practice opportunity, the student is assumed to master the skill with some probability. The BKT model is parametrized by the following four parameters:

- $P(L_0) = P(K_1 = 1)$: the initial probability of knowing the skill (before the student is given any practice opportunities)
- $P(T) = P(K_{i+1} = 1|K_i = 0)$: the probability of mastering a skill at each practice opportunity (if the student has not yet mastered the skill)
- $P(G) = P(C_i = 1|K_i = 0)$: the probability of guessing
- $P(S) = P(C_i = 0|K_i = 1)$: the probability of "slipping" (answering incorrectly despite having mastered the skill)

3. IDENTIFIABILITY

In their 2007 paper, Beck and Chang claimed that BKT is not identifiable, illustrating this with a particular example of three different BKT models [7]. For concreteness we include these models in Table 1. The authors consider the case of predicting the probability of correctness under these three models as the students receive practice opportunities, but in absence of any observation about the student's performance. They use plots as in Figure 1 to claim that the three models make very different predictions about student knowledge (Figure 1 (a)), but make identical predictions about student performance (Figure 1 (b)). They claim,

> All three of the sets of parameters instantiate a knowledge tracing model that fit the observed data equally well; statistically there is no justification for preferring one model over another. This problem of multiple (differing) sets of parameter values that make identical predictions is known as identifiability.

However, this is not correct since no data was used to fit these curves; the curves are predicting the probability that a student will know the skill or will answer the skill correctly at each practice opportunity i, *when we have no prior performance or data on the student.* In order to take past data from a student into account, we actually want to predict $P(K_i = 1|C_1, \ldots C_{i-1})$ and $P(C_i = 1|C_1, \ldots C_{i-1})$ and this is indeed what we do in practice when doing knowledge tracing; we make predictions based on our past observations. Figure 2 shows the curves predicting these conditional probabilities for a particular sequence of correct/incorrect answers for a student (namely we use $(1, 0, 0, 0, 0, 0, 0, 1, 1)$). We find that even when we condition on a single observation (i.e., for $P(C_2 = 1|C_1)$), the three models make vastly different predictions, and as we collect more data, the models continue to make very different predictions. In fact, except for $P(C_1 = 1)$, the models never agree on the probability that a student would answer the step correctly.

Formally, a model is said to be *identifiable* if there are no two distinct sets of model parameters θ and θ' that can give rise to the same joint probability distribution over observations under that model. As far as inference is concerned, identifiability means that the likelihood function of the model has only one global maximum, so inference of the true model parameters is possible. In the case of BKT, the model would be identifiable if for any two distinct sets of BKT parameters, θ and θ',

$$P_\theta(C_1, C_2, \ldots, C_n) \neq P_{\theta'}(C_1, C_2, \ldots, C_n)$$

for some $n \geq 1$. What Beck and Chang show is that there can be infinitely many models that share the same set of marginal distributions $P(C_1), P(C_2), \ldots, P(C_n)$. This does not mean the model is unidentifiable. As we saw from Figure 2, the conditional distribution $P(C_n|C_1, \ldots, C_{n-1})$ is quite different for each model, and so the joint distribution $P(C_1, \ldots, C_n)$ is also very different for the three models.

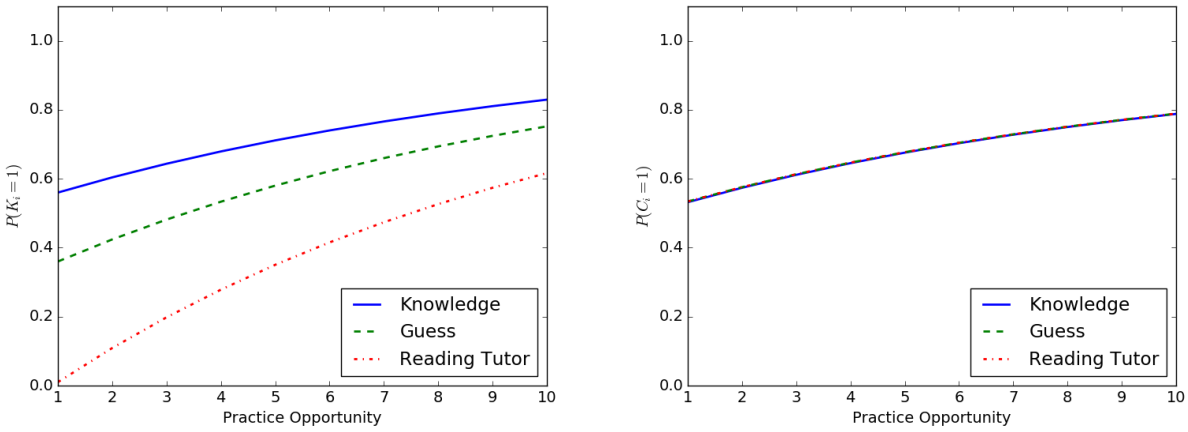
It turns out there has been a substantial amount of work, going back 50 years and continuing to this day, on finding the conditions for which hidden Markov models are identifiable [15, 1, 2, 17, 10]. Although much of the literature focuses on particular types of HMMs (e.g., stationary, irreducible) that do not include the standard BKT model, Anandkumar et al. have recently shown that, subject to some non-degeneracy conditions, a large class of HMMs, which includes BKTs, is identifiable with just the joint probability distributions for up to three sequential observations [4]. That is, knowing $P(C_1)$, $P(C_1, C_2)$, and $P(C_1, C_2, C_3)$ is enough to infer the unique model parameters, subject to non-degeneracy conditions. In our context, the conditions are that $P(L_0) \notin \{0, 1\}$, $P(T) \neq 1$, and $P(G) \neq 1 - P(S)$. This suggests that as long as we have more than two observations per student, BKT models with reasonable parameters are identifiable and there is a single global maximum to the likelihood function. Feng recently independently showed the same result directly for BKT models, except without requiring the condition that $P(L_0) \neq 0$ [9]. One advantage of relying on general identifiability results for HMMs is that we can use the same results to show the conditions under which related student models that can also be modeled as HMMs are identifiable[1].

This misuse of the term "identifiability" has lead to multiple subsequent papers in the educational data mining community throughout the past decade which have similarly given a mistaken description of the underlying phenomena [5, 16, 13, 12]. Two papers, however, have correctly identified that the

[1]For example, for the BKT model with forgetting, where $P(F) = P(K_{i+1} = 0|K_i = 1) \neq 0$, we can show that the model is identifiable with the same conditions, except that we require $P(T) \neq 1 - P(F)$ instead of $P(T) \neq 1$. We can also easily show the conditions under which multi-state extensions of BKT such as the model introduced in Section 4.2 are identifiable. These conditions can be derived from Condition 3.1 and Proposition 4.2 of [4]. See also the note under Proposition 3.4 of [3].

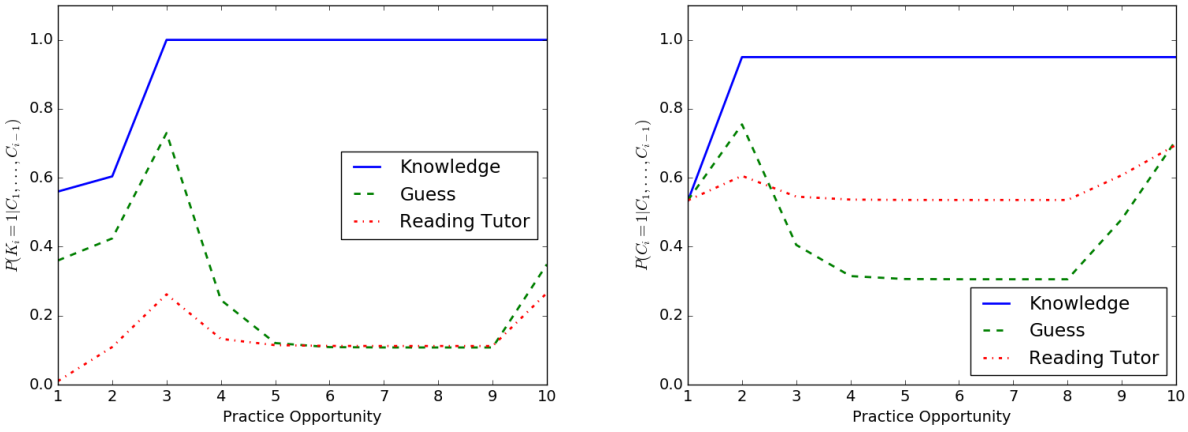
	Model		
Parameter	Knowledge	Guess	Reading Tutor
$P(L_0)$	0.56	0.36	0.01
$P(T)$	0.1	0.1	0.1
$P(G)$	0	0.3	0.53
$P(S)$	0.05	0.05	0.05

Table 1: The three BKT models used by Beck and Chang [7] to claim BKT is unidentifiable. The models are chosen to have very different semantic interpretations. The Knowledge model requires the student to master the skill to get it correct, the guess model relies on the student guessing, and the Reading Tutor model has an even higher probability of guessing, but it was based on models actually used by the Reading Tutor [14].



(a) Learning Curve (b) Performance Curve

Figure 1: Hypothetical learning and performance curves for three models from [7], in absence of any data.



(a) Learning Curve (b) Performance Curve

Figure 2: Learning and performance curves for three models from [7] conditioned on all past observations for a student whose observed trajectory is as follows: $(C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9) = (1, 0, 0, 0, 0, 0, 0, 1, 1)$

"identifiability problem" is limited to the case where there is no data [18, 11]. Even though this is not a statistically precise claim, it does show that some researchers have the correct understanding behind the phenomenon. Van de Sande distinguishes between the two cases where predictions are made in the absence of data and where they are made in the presence of data, and claims that the source of the identifiability problem in the former case is that the predictions can be completely determined by three parameters, so there is a degree of freedom [18]. When we are making predictions, however he claims there is no identifiability problem, because $P(K_i|C_i)$ depends on four parameters [18]. While he has correctly identified the absence of an identifiability problem in the presence of data, we believe that there is still confusion about the identifiability problem in the community (e.g., some of the papers that show a misunderstanding of the issue are more recent than [18]). We hope to make the absence of an identifiability problem more clear and elucidate the phenomena and misconceptions surrounding it. Gweon et al. also distinguish between two cases which they refer to as the BKT model without measurement and the BKT model with measurement, and show, as van de Sande did, that the former depends on three parameters (hence the "identifiability problem") whereas the latter depends on all four [11]. However, they claim this does not necessarily mean that the BKT model with measurement does not suffer from an identifiability problem, and actually claim that it still does suffer from an identifiability problem, because empirically, they found that for some data, fitting BKT models many times resulted in a wide spread of possible parameters [11]. However, this cannot be due to the presence of an multiple global maxima, which we have shown cannot exist, and hence must be due to multiple local optima.

The work closest to ours is Feng's recently published dissertation [9]. The author gives a similar explanation to ours for why Beck and Chang's claim was incorrect and also proves that the BKT model is identifiable directly [9]. However, we believe the exposition there is perhaps less accessible to the educational data mining community and will likely not obtain the visibility needed to clear the misunderstandings surrounding the identifiability of BKT. In this paper, we not only focus on identifying the misidentified identifiability problem, but also understanding the confusion surrounding it as well as pointing out actual issues with fitting BKT models that have been conflated with identifiability. This is the focus of the rest of the paper.

There are three potential sources of confusion that we believe could be and have been misconstrued as an identifiability problem:

1. *A priori predictions.* That multiple models, which make very different claims about student's knowledge state over time, could predict the same probability that students answer questions correctly over time *in the absence of data.* This is the problem that Beck and Chang conflated with identifiability, and many researchers thereafter also treated as identifiability. As we showed above, van de Sande, Gweon et al. and Feng correctly identified what is happening here [18, 11, 9].

2. *Multiple local optima.* It is well known that the expectation-maximization algorithm that is commonly used to fit BKT models is suceptible to converging to local optima of the likelihood function rather than converging to the global optimum. While Beck and Chang clearly did not conflate this with the identifiability issue, we saw that other researchers such as Gweon et al. have possibly conflated the two. In order to avoid local optima, one can use a grid search over the entire parameter space or run multiple iterations of the expectation-maximization algorithm with different initializations of the parameters.

3. *Semantic model degeneracy.* Baker et al. identified another problem with BKT models, which they termed model degeneracy [5]. A model is said to be semantically degenerate[2] when it is inconsistent with the conceptual assumptions underlying the BKT model. The problem is when the model that best fits our data is semantically degenerate. Even though Baker et al. clearly contrasted this to the (supposed) identifiability problem, we claim that this is the problem that Beck and Chang attempted to fix in their paper. We will now focus on better understanding this problem.

4. SEMANTIC MODEL DEGENERACY

In their paper, Beck and Chang propose a way to get around the identifiability problem. They propose using Dirichlet priors to encode prior beliefs about the BKT parameters, which will in turn bias the model search towards more reasonable parameters [7]. They motivate their method as follows:

> We have more knowledge about student learning than the data we use to train our models. As cognitive scientists, we have some notion of what learning "looks like." For example, if a model suggest that a skill gets worse with practice, it is likely the problem is with the modeling approach, not that the students are actually getting less knowledgeable. The question is how can we encode these prior beliefs about learning?

The problem they appear to be describing is that some models have parameters that do not match our intuitions of student learning, i.e., they are exactly describing the issue of semantic model degeneracy (and not that of unidentifiability). Baker et al. later provide another solution to tackling semantic model degeneracy by using contextual features to estimate the guess and slip parameters [5]; however, interestingly they did not view Beck and Chang's original solution as a way of tackling semantic model degeneracy, treating it as a way to tackle identifiability as the authors originally claimed.

Having shown that identifiability is not an issue with BKT, and given that there are easy ways to tackle the existence of local optima, we believe semantic model degeneracy is perhaps the most important problem with respect to fitting BKT models that needs to be better understood and tackled. Essentially, the problem arises because the BKT is simply a

[2]We refer to this property as semantic model degeneracy to distinguish it from mathematically degenerate parameters that would result in BKT models being unidentifiable, as described above.

particular form of a two-state hidden Markov model and it will try to fit the best two state hidden Markov model it can to the data; our model fitting procedures do not understand that the $K_i = 1$ state is supposed to correspond to mastering a skill, and so it might fit a model that does not match our intuitions of mastery. We will try to understand this in more detail below, but first we aim to characterize the types of semantic model degeneracy that have been pointed out in the literature.

4.1 Types of Semantic Model Degeneracy

Baker et al. distinguish between two forms of semantic model degeneracy: *theoretical degeneracy* and *empirical degeneracy* [5]. They define a model to be theoretically degenerate when either the guess or the slip parameter is greater than 0.5. They define a model to be empirically degenerate if one of two things occur: (1) for some large enough n the model's estimate of the student having mastered the skill decreases after the student gets the first n skills correct or (2) for some large enough m, the student does not achieve mastery (our estimate of the student having mastered the skill does not go beyond 0.95) even after m consecutive correct responses [5]. The authors arbitrarily chose the values $n = 3$ and $m = 10$. Note that the first form of empirical degeneracy is only possible if $1 - P(S) < P(G)$ (i.e., the student is more likely to answer a question correctly if they have not mastered a skill than if they have mastered a skill), as was shown by van de Sande [18]. This is true, even for $n = 1$. Thus, this first notion of empirical degeneracy is equivalent to $P(G) + P(S) > 1$, which implies either $P(S) > 0.5$ or $P(G) > 0.5$, meaning that it always implies theoretical degeneracy! Huang et al. have noted that while $P(G) + P(S) > 1$ definitely implies semantically degenerate parameters as it contradicts mastery, the condition that $P(G) < 0.5$ and $P(S) < 0.5$ may not always be necessary for the parameters to be semantically meaningful, since, for example, there may be some domains where the student can guess the correct answer easily [12]. We agree that suggesting $P(G) < 0.5$ is degenerate does seem somewhat arbitrary depending on the domain; however, we do think $P(S) > 0.5$ should be characterized as a form semantic degeneracy, because, as Baker et al. claimed, it does not make sense for a student who has mastered a skill to answer questions of that skill incorrectly most of the time—that goes against our intuitions of what mastery means. In any case, it does not seem like the distinction between theoretical and empirical degeneracy is a clear one, so we suggest categorizing the forms of semantic model degeneracy by what they suggest about student learning:

- *Forgetting*: This is a result of $P(G) + P(S) > 1$, which suggests that not only are students not learning, but that students have some probability of losing their knowledge over time. Another way to view this degeneracy is that the state we would conceptually call the mastery state is now the state where performance is worse.
- *Low Performance Mastery*: This is a result of $P(S) > 0.5$. Alternatively, we can set our threshold for low performance mastery to be lower (e.g., P(S) > 0.4).
- *High Performance Guessing*: This is a result of $P(G) > t$, where t is some threshold. As mentioned earlier, this seems like a weak form of degeneracy, as students can often guess an answer easily even if they have not mastered a skill, but we can set t to a large enough value, to make this a form of model degeneracy.
- *High Performance $\not\Rightarrow$ Learning*: This is the second form of empirical degeneracy given by Baker et al. [5]: for some choice of m, the probability that the student has achieved mastery is less than some threshold p (typically taken to be 0.95) after m consecutive correct responses

4.2 Sources of Semantic Model Degeneracy

We will now consider a possible explanation for why BKT models are so prone to semantic model degeneracy (which we believe to be part of the reason that researchers look towards identifiability and local optima to explain the strange parameters that result from fitting BKT models). First of all, note that forgetting degeneracy will occur whenever students actually do forget or when they learn misconceptions; it is not unreasonable to believe that students will sometimes learn and reinforce a misconception, causing their knowledge of some skill to decrease over time. Thus, while this form of degeneracy technically violates our notion of mastery, it is to be expected if we switch the semantic interpretation of the two states and suppose that students forget instead of learn. We now consider sources of the other forms of semantic model degeneracy. We claim that such forms of semantic model degeneracy can result from not accurately being able to capture the complexity of student learning with a two state HMM. When this is the case, fitting the data with a two state HMM will result in trying to find the best fit of the data for a two state HMM, and not to come up with a model that tries to accurately model the data while also matching our intuitions about what it means for a student to have mastered a skill.

To support our claim, suppose student learning is actually governed by a 10-state HMM with ten consecutive states representing different *levels* of mastery. From each state, the student has some probability of transitioning to the next state (slightly increasing in mastery), and from each state, the student has a probability of answering questions correctly, and this probability strictly increases as the student's level of mastery increases. Specifically consider the model presented in Table 2. Now suppose we try to use a standard BKT model to fit data generated from this alternative model of student learning. The first two columns of Table 3 show the parameters of BKT models fit to 500 sequences of 20 practice opportunities or 100 sequences of 200 practice opportunities, both generated from the the model in Table 2. Notice that the model fits (nearly) degenerate parameters in both cases. When we only have 20 observations per student, the model estimates a very high slip parameter; this is because it has to somehow aggregate the different latent states which correspond to different levels of mastery, and since not many students would have reached the highest levels of mastery in 20 steps, it is going to predict that students who have "mastered" the skill are often getting it wrong. However, what's more interesting is that for the same model, if we simply increase the number of observations per student from 20 to 200, we find that the slip parameter is reasonably small, but now the guess probability is 0.49! This is because, by

	State i									
Parameter	0	1	2	3	4	5	6	7	8	9
$P(K_0 = k)$	0.1	0.1	0.1	0.2	0.2	0.3	0	0	0	0
$P(C_i = 1 \mid K_i = k)$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$P(K_i = k+1 \mid K_i = k)$	0.4	0.3	0.2	0.1	0.05	0.05	0.05	0.05	0.05	-

Table 2: Alternative model of student learning where there are ten levels of mastery.

	10-State HMM		AFM	
Parameter	20	200	20	200
$P(L_0)$	0.30	0.001	0.09	0.001
$P(T)$	0.05	0.02	0.05	0.05
$P(G)$	0.27	0.49	0.14	0.28
$P(S)$	0.44	0.13	0.46	0.03

Table 3: BKT models fit to data generated from the model described in Figure 2 and an additive factors model described in the text. The first column for each model is fit to 500 sequences of 20 practice opportunities, while the second column is fit to 100 sequences of 200 practice opportunities. The models were fit using brute-force grid search over the entire parameter space in 0.01 increments for the parameters using the BKT Brute Force model fitting code [6].

this point most students have actually reached the highest level of mastery, so to compensate for the varying levels of mastery that occurred earlier in student trajectories, the model will have to estimate a high guess parameter. So we find that not only can alternative models of student learning lead to fitting (near) degenerate parameters, but varying the number of observations can lead to different forms of degeneracy! This is a counterintuitive phenomenon that we believe is not the result of not having enough data (students) to fit the models well, but rather the result of the mismatch between the true form of student learning and the model we are using the fit student learning.

We find similar results if we fit a BKT model to data generated from another alternative model of student learning that is commonly used in the educational data mining community, the additive factors model (AFM) [8]. In particular, we used the model

$$P(C_i = 1) = \frac{1}{1 + exp(-\theta + 2 - 0.1i)}$$

where $\theta \sim \mathcal{N}(0, 1)$ is the student's ability[3]. The second two columns of Table 3 show the parameters of BKT models fit to data generated from this model. We again find that when using only data with 20 practice opportunities, we fit a high slip parameter, but when we using data with 200 practice opportunities, we fit a higher guess parameter and a very small slip parameter.

Additionally, notice that for the parameters fit to the 10-state HMM, the probability of transitioning to mastery is very small when we fit to sequences with 200 practice opportunities. Since the transition probability is small and the guess probability is large, we also have high performance $\not\Rightarrow$ learning degeneracy for this model for $m = 10$. That is,

$$P(K_{11} = 1 \mid C_1 = 1, C_2 = 1, \ldots, C_{10} = 1) \approx .89 < 0.95$$

This is yet another form of degeneracy that does not exist in the model fit to sequences of 20 practice opportunities. Furthermore, notice that when we have 200 observations, the probability of transitioning to mastery is smaller than $P(K_i = k + 1 \mid K_i = k)$ for all states i in the model that generated the data (Table 2). Again, this is because the best fitting BKT model will aggregate low performing states and high performing states, so a single transition in the BKT model between these two aggregate states will have to loosely correspond to the student transitioning several times in the actual 10-state HMM. Thus, while the learned BKT model makes it appear as though learning happens very slowly, according to the true student model, learning actually occurs much more often but in more progressive increments. This suggests that if we use some automated technique to detect if a skill is useful for student learning, we may conclude it is not, if we do not allow for the possibility that students are learning progressively.

These observations have important implications for how learned models can be used in practice. Using such a BKT model to predict student mastery can lead to problematic inferences. For example, for the first model in Table 3, the BKT model assumes that when a student has reached mastery, they have a 56% chance of answering a question correctly, whereas a student who has actually mastered the skill will have a 90% chance of answering correctly (see Table 2). Thus, an intelligent tutoring system that uses such a BKT model to determine when a student has had sufficient practice on a problem, will likely give far fewer problems to the student than they actually need in order to reach mastery!

There are several potential ways that future work can proceed in light of these findings. One is that we should be giving our model fitting procedures more domain knowledge about the kind of model we want it to fit. This is essentially what Beck and Chang did by using Dirichlet priors [7] and what Baker et al. did by estimating the guess and slip parameters using context [5]. But perhaps there are other ways of doing this where we do not need to give context-dependent domain knowledge to the model per se, but rather come up with a model that realizes the difference between a student having mastered a skill or not (which the BKT model cannot do). However, this may not be ideal in some cases where student learning cannot accurately be modeled by BKT with semantically plausible parameters. For example when we have forgetting degeneracy, we should probably not force

[3]This model suggests that students who are two standard deviations above the mean initially will answer correctly half the time, and after 20 practice opportunities the average student will answer correctly half the time.

the parameters to suggest learning is occurring when it may not be. Another way to proceed is to consider alternative student models, which is an active area of educational data mining research. Perhaps, obtaining semantically degenerate parameters from a fit should signal that our students may be learning in more complicated ways than the simple BKT model can predict, and so we should try to find alternative models that fit our data better without yielding semantically degenerate parameters. Finally, even if our model is semantically degenerate, it does not necessarily make the BKT model useless. The result of fitting a BKT model is that we get the best fit of the data given that we are modeling the data with a two-state HMM (if we disregard local optima). Presumably, such a model can give us some insights about student learning even if it is not modeling student mastery. So perhaps we can use such semantically degenerate models to understand student learning rather than to predict student mastery.

5. CONCLUSION

We have explored the issues of identifiability and semantic model degeneracy in Bayesian Knowledge Tracing. We have shown that what researchers posited was an identifiability problem is actually not an identifiability problem, and by using a result from the literature on learning hidden Markov models, we showed that an identifiability problem does not exist for BKT models (with the exception of some mathematically degenerate cases that should not come up in practice). We then examined the various issues with fitting BKT models that have been conflated with identifiability. We offered what we believe to be new insights on one potential source of semantic model degeneracy. We believe analyzing the sources of semantic model degeneracy in more detail can be a fruitful direction for future research. For example, it could be useful to know what BKT parameters result from fitting various other popular models of student learning. It would also be informative to see if we can find automated ways of detecting which assumptions of BKT are not met in our data (e.g., the number of levels of mastery, the independence of different skills). Such analyses could help in devising better student models, and ultimately may lead to a better understanding of student learning.

6. ACKNOWLEDGEMENTS

The research reported here was supported, in whole or in part, by the Institute of Education Sciences, U.S. Department of Education, through Grants R305A130215 and R305B150008 to Carnegie Mellon University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Dept. of Education.

7. REFERENCES

[1] E. S. Allman, C. Matias, and J. A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, pages 3099–3132, 2009.

[2] Y. An, Y. Hu, J. Hopkins, and M. Shum. Identifiability and inference of hidden markov models. Technical report, Technical report, 2013.

[3] A. Anandkumar, R. Ge, D. J. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[4] A. Anandkumar, D. Hsu, and S. M. Kakade. A method of moments for mixture models and hidden markov models. In *Conference on Learning Theory*, pages 33–1, 2012.

[5] R. S. Baker, A. T. Corbett, and V. Aleven. Improving contextual models of guessing and slipping with a truncated training set. *Human-Computer Interaction Institute*, page 17, 2008.

[6] R. S. Baker, A. T. Corbett, S. M. Gowda, A. Z. Wagner, B. A. MacLaren, L. R. Kauffman, A. P. Mitchell, and S. Giguere. Contextual slip and prediction of student performance after use of an intelligent tutor. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 52–63. Springer, 2010.

[7] J. E. Beck and K.-m. Chang. Identifiability: A fundamental problem of student modeling. In *International Conference on User Modeling*, pages 137–146. Springer, 2007.

[8] H. Cen. *Generalized learning factors analysis: improving cognitive models with machine learning*. Carnegie Mellon University, 2009.

[9] J. Feng. *Essays on learning through practice*. PhD thesis, The University of Chicago, 2017.

[10] É. Gassiat, A. Cleynen, and S. Robin. Inference in finite state space non parametric hidden markov models and applications. *Statistics and Computing*, 26(1-2):61–71, 2016.

[11] G.-H. Gweon, H.-S. Lee, C. Dorsey, R. Tinker, W. Finzer, and D. Damelin. Tracking student progress in a game-like learning environment with a monte carlo bayesian knowledge tracing model. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 166–170. ACM, 2015.

[12] Y. Huang, J. Gonzalez-Brenes, R. Kumar, and P. Brusilovsky. A framework for multifaceted evaluation of student models. In *Proceedings of the 8th International Conference on Educational Data Mining*. University of Pittsburgh, 2015.

[13] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. *International Educational Data Mining Society*, 2012.

[14] J. Mostow and G. Aist. Smart machines in education. chapter Evaluating Tutors That Listen: An Overview of Project LISTEN, pages 169–234. MIT Press, Cambridge, MA, USA, 2001.

[15] T. Petrie. Probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 40(1):97–115, 1969.

[16] S. Ritter, T. K. Harris, T. Nixon, D. Dickison, R. C. Murray, and B. Towle. Reducing the knowledge tracing space. *International Working Group on Educational Data Mining*, 2009.

[17] P. Tune, H. X. Nguyen, and M. Roughan. Hidden markov model identifiability via tensors. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2299–2303. IEEE, 2013.

[18] B. van De Sande. Properties of the bayesian knowledge tracing model. *JEDM-Journal of Educational Data Mining*, 5(2):1–10, 2013.