How productive are homework and elective practice? Applying a post hoc modeling of student knowledge in a large, introductory computing course

Max Fowler
University of Illinois
Urbana, IL, USA
mfowler5@illinois.edu

Binglin Chen
University of Illinois
Urbana, IL, USA
chen386@illinois.edu

Matthew West
University of Illinois
Urbana, IL, USA
mwest@illinois.edu

Craig Zilles
University of Illinois
Urbana, IL, USA
zilles@illinois.edu

ABSTRACT

In this paper, we attempt to estimate how much learning happens in required practice activities (homework) relative to elective practice activities (studying). This analysis is done in the context of a large enrollment ($N = 601$) introductory programming course that made heavy use of autograding randomizing question (item) generators. Because these item generators (and other problems) were used as homework, on practice exams, and as part of exams, a given student may have encountered the same generator multiple times during the class, providing snapshots of the evolution of the student's ability to complete that problem correctly.

We use a *post hoc* model of "this-item-correct" prediction to estimate individual student knowledge on each attempt of a given question. Across five exams, correctness tracing attributes 57-65% of the learning that occurs to the homework period and the remainder to elective practice (the study period).

Keywords

assessment; CS1; exams; student learning; homework

1. INTRODUCTION

A well-designed course provides students with many opportunities to learn (e.g., readings, direct instruction, activities with peers, homework). While summative assessment allows us to estimate how much learning has occurred, it doesn't shed light on where the learning happened. If we could attribute learning to the activities in which it occurred, this would allow teachers to increase their use of effective activities and deprecate ineffective ones. Our goal as educators is to engage students, to assist both them and us in diagnosing their progress, and to provide formative experiences during their learning careers [8, 15, 21, 34].

In most courses, the bulk of the students' time is spent outside of course meetings, either completing homework or performing elective practice (studying). It has been shown that well-formed homework has a positive impact on student performance and motivation [5, 6, 14, 22]. There are, however, disagreements between experts among the learning and assessment communities on how to craft good homework [2, 37]. Studying is usually motivated by a desire to score well on exams and does not typically have a grade associated with it [33].

We were curious to explore the degree to which we can attribute student learning between two kinds of formative practice activities: required homework and elective practice performed prior to a summative assessment. Additionally, as our course utilizes multiple types of questions, we were curious to know if student experiences differed between types. To do so after the completion of the course, we use a *post hoc* knowledge estimation method developed by Chen et al [9]. This method, which we call "correctness tracing" (CT) as shorthand, models student learning as the likelihood of students getting specific questions correct on a given attempt for those questions. The method estimates the chance of a student getting "this-item-correct" for a given item (question) at every attempt the student makes on that item, for all items.

We apply CT to student submission data from an on-campus introductory programming course. The course used randomly selected questions from question pools and random item generators for exam creation, with many of the questions appearing previously on homework (and optional practice exams) as a studying motivator for students. We use data from student homework, practice exams, and these proctored exams to build a cohesive snapshot of student experience with the same questions in multiple contexts. Specifically, we share our experience investigating students' learning in this fashion to address the following questions:

RQ1: How much learning happens during required practice activities (homework) relative to elective practice (studying)?

RQ2: Does student learning differ based on the type of the questions (e.g., multiple-choice vs. short answer) asked?

The rest of our paper is organized as follows. Section 2 describes related work on student learning and knowledge tracing. Section 3 discusses the course from which we collected data and the handling of that data. In Section 4, we explain the assumptions behind CT and detail our use of the method. We follow with our results from the modeling in Section 5 and with interpretation and limitations in Section 6. We conclude in Section 7.

2. RELATED WORK

2.1 How are students learning on homework and through studying?

How students learn is an area of significant study. We are specifically interested here in how formative assessment (e.g. homework) helps students learn. Historically, formative assessment is claimed to benefit student learning, although there is little consensus on what exactly makes good formative assessment [7]. There is evidence, however, that frequent and distributed practice, such as frequent testing, boosts student achievement and learning [1, 4, 24, 32].

Research on homework often considers benefits to students' motivation and self-regulatory ability as opposed to just content learning. Ramdass and Zimmerman used correlational studies to show that homework leads to higher self-regulatory abilities and traits, like time management and self-efficacy [29]. Similarly, Bembenutty and White showed that students who approach homework with help-seeking attitudes and as motivating exercises displayed stronger academic performance [5].

Mandatory homework is found to be beneficial in existing research, but in large part due to feedback. Gutarts and Bains found that homework that provides feedback appears to enhance student performance [14]. However, Johnson and McKenzie found that while mandatory homework may incentivize homework-related motivation and learning, it was not correlated with exam performance in their macroeconomics course [17]. Ryan and Hemmes found homework was correlated with improved quiz performance, but that points are a necessary contingency to get students to do homework, with feedback-only approaches reducing student engagement [31].

The benefits of studying are less clearly defined. Chew suggests the benefit of study can be improved by teaching students *how* to study and that expecting students to know how without designing assignments and material to aid their studying may be a mistake on the part of some instructors [11]. Fakcharoenphol et al. found that there was a learning increase in studying old exams with solutions and feedback, but that this learning may be shallow [13].

The idea that studying itself may be comparatively shallow is supported in the literature on long-term retention. Karpicke and Blunt found that the retrieval practice from exams was superior for learning than elaborative studying processes [18]. Additionally, Roediger and Nestojko found that, while studying did improve long-term retention of concepts, retrieval during testing still had superior results [30].

2.2 Knowledge tracing and student modeling

There is a wealth of work on different methods of tracing student knowledge and modeling student learning and student behavior. Many of these stem from Corbett and Anderson's original knowledge tracing paper [12]. Since the original tracing paper, there has been more work on dealing with issues such as student slip and guess behavior, the benefits and traceability of learning resources, and other parts of students' learning environments. Pelanek's significant review shows how learner modeling has grown to encompass domain knowledge structuring, learner clustering, student observations, and more just over the last decade [27]. We address a few below.

Pardos and Heffernan modeled individualized learning in Bayesian knowledge tracing (BKT) [25]. In their method, students' skills were used to set each student's individualized knowledge for more accurate individual knowledge tracing. They later introduced individual item difficulty as a way to make knowledge tracing more robust to unseen items [26]. As opposed to skills being used for individual student priors, Khajah et al. used latent factors pulled from student populations to predict individual student performance [20]. Other approaches use machine learning methods to estimate student guess or slip chances as opposed to students having not yet learned course material [3].

Deep learning methods have also been applied to knowledge tracing in deep knowledge tracing (DKT) [28]. Additions to DKT include prerequisite modeling in students' concepts [10], problem level features like time to complete and student hint usage [39], and dynamic student grouping based on performance [23]. There is some evidence to suggest that, while DKT is powerful, BKT can similarly be extended and that the gains do not require "deep" learning techniques explicitly [19]. Additionally, methods such as predictive failure analysis can perform similarly to DKT so long as care is taken to structure data appropriately [38].

3. DATA COLLECTION

Our data was collected in a large enrollment, introductory programming course for non-CS majors in Fall 2019. The course had 601 total students, with 246 women and 355 men. The majority of students who took the course were freshmen (67%) and sophomores (21%). The course predominantly taught Python programming with some coverage of basic Excel and HTML/web concepts.

3.1 Course context

The course was organized as a flipped class that covered one major topic each week. Students were expected to complete readings in an interactive textbook and an assignment consisting of true/false and multiple-choice questions prior to

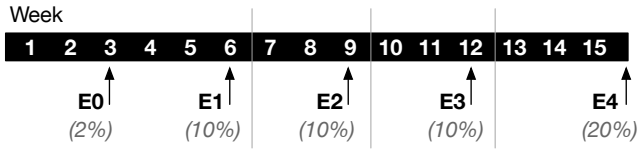

Figure 1: Every three weeks the course had a proctored exam (E0 to E4). Weight relative to the final course grade is provided as a percentage.

lecture. The weekly 90-minute lecture used peer instruction to reinforce concepts, and the weekly 80-minute lab consisted of practice activities students could complete individually or in pairs, supervised by course staff. Finally, each topic culminated with a weekly homework assignment that consisted of a mix of short answer (e.g., "What is the value of the variable x after the following piece of code executes?", "Write a statement that removes the 4th element of a list called 'animals'.") and small programming (i.e., no more than a small function) questions.

Due to the size of the course, almost all of the homework activities were auto-graded. The course used the open-source assessment platform (PrairieLearn) [35, 36] for all homework and other assessments. PrairieLearn both instantly grades student submissions and provides automatic feedback. Homework assignments were configured for students to be fearless: there was no penalty for wrong answers, only points to gain as they got answers correct. On homework, this allowed students to practice with course content repeatedly until they got the correct answer. Students were able to repeat questions until they earned full credit and revisit questions at any point for studying purposes.

Many of the homework questions were *item generators* that could produce many possible questions of similar difficulty on the same topic [16]. The true/false and multiple-choice item generators randomly selected items from pre-populated pools of questions. Short answer questions are randomly parameterized (e.g., changing the list a student has to read or changing the method applied to a given list). To encourage mastery, homework often expected students to correctly answer these item generators multiple times. Weekly homework assignments typically included 12 to 30 items or item generators and students needed to complete 90% of them to achieve a full score on the homework.

The course's primary mean of summative assessment was through five proctored exams. All the exams had a 50-minute fixed time limit, except for the final exam (E4) which allowed for 3 hours. All but the first exam were worth a significant portion ($\geq$ 10%) of the course grade. These exams were conducted in a proctored computer lab with student scheduled exam times within a three-day window [40–42]. Students were given access to a Python interpreter and Python's documentation, but no other resources were provided. The exam schedule is given in Figure 1.

Exams featured all four kinds of questions seen on homework (T/F, MC, short, programming), except for E0 which did not have programming questions. Each exam consisted of 20–30 question slots (41 on the final). Each slot drew randomly from a pool of questions on a given topic with similar difficulties. Most questions permitted students to attempt them multiple times with a score penalty for each subsequent incorrect attempt until chances to earn credit were exhausted.

Because of the course's heavy use of item generators and to motivate students to take homework seriously, a significant fraction of the exams were drawn from the course's pre-lecture and homework assignments. In general, 85–90% of the pools on the exam were drawn from questions previously on homework, and exam-only "hidden" questions were written with similar form and content to previous homework questions. Prior to each exam, students were provided access to a practice exam generator that was similar to the actual exam generator, but without the hidden questions. Reused programming questions are largely recall exercises, as most do not feature random generation. Short answer questions are transfer tasks as they are all parameterized and no two instances of the question should possess the same exact parameters and the same expected student answer.

In spite of the exams including a large fraction of previously seen material, we don't believe that rote memorization was a useful strategy for these exams due to their heavy use of randomization and question pools combined with a large number of questions (20–30) on the exam. True/false and multiple-choice slots on the exam generally drew from pools of 20 to 100 questions, while short answer and programming question slots had pool sizes of 5 to 12. In addition, short answer item generators typically produce at least dozens of meaningfully different variants.

3.2 Homework and study periods

The decision for exams to *mostly* use the same questions as homework assignments and practice generators created an interesting context for attributing student learning. Specifically, we could analyze student performance on homework assignments, practice exams, and actual exams to observe how students' ability to answer these questions improved as they engaged with course material. We pulled all student submissions from PrairieLearn for the entire semester, keeping only submissions for any questions that appeared on both homework and exams.

We cleaned this data set by removing students who had not completed all of the exams, retaining 584 of the 601 students. In total, we retained 1,064,547 individual submissions across homework, optional practice, and exams. Each submission's score ranges from 0 (incorrect) to 1 (full credit), with scores in-between indicating partial credit.

We subdivide our analysis of the course by exam, focusing on the three week window preceding each of the five exams. As shown in Figure 2, each exam is comprehensive, including material that was present on previous exams. For this analysis, we focus solely on the content introduced since the previous exam to see how practice during the homework and study periods contribute to learning for the material's first summative assessment.

Each student submission is assigned to one of three periods: homework, study, and exam (Figure 3):

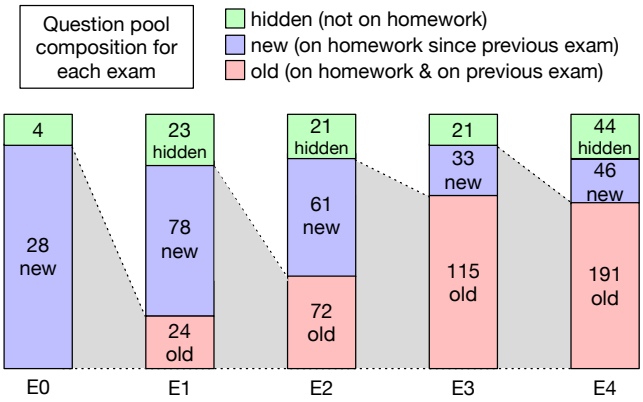

Figure 2: Exams are cumulative and largely drawn from item generators and questions previously on homework. *Hidden* questions only appear on exams. *New* questions were on homework since the previous exam, while *old* questions were previously on earlier homework and one or more previous exams. While the fraction of exam slots dedicated to *old* questions does increase as the semester progresses, this figure is somewhat deceptive because *old* pools typically have many more questions than *new* and *hidden* pools, except on the final (E4) where each week's material is represented equally.

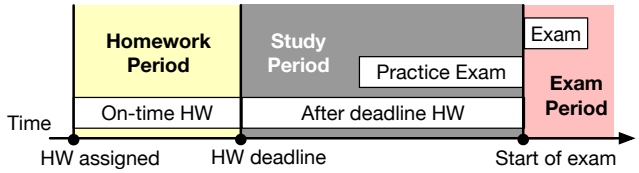

Figure 3: We subdivide the students' practice into two periods: the *homework* period is all homework attempts before the deadline. The *study* period is all attempts on practice exams and any homework attempts after the deadline.

- The *homework* period includes all the submissions to homework on or before the homework due date. Submissions in this period represents required practice; while students are allowed as many submissions as they need to get full credit, there is a deadline to receive that credit.
- The *study* period includes all submissions on practice exam generators as well as any submissions on homework after the homework deadline. The homework system remains open and students can repeat problems and complete any problems not previously completed (only 90% of questions are needed to achieve a full homework score). Submissions in this period are elective practice, bearing no credit directly.
- The *exam* period includes the submissions on the actual exam.

The above periods are coarsely defined to capture the difference between the time spent on required practice with homework assignment and any additional practice following the homework deadline. For our context, problems being completed by students on practice exams as well as after a homework deadline are both elective activities and are suitable to be counted together.

For our analysis, we also tag each student's *first* attempt on each question on homework, so that we can estimate the student's ability to solve that question gained before attempting the question the first time (e.g., from readings, lecture, or solving other problems). A breakdown of the number of submissions during each period is provided in Figure 4. The decrease in submissions throughout the semester in the homework and studying buckets is a result of homework shifting toward fewer, more difficult problems as the semester progresses.

4. METHODS

To analyze the evolution of student knowledge from homework to exam time, we track student learning at the granularity of individual item generators. This is clearly a significant approximation to reality for two reasons: 1) because of pools (of true/false and multiple-choice questions) and parameter randomization (for short answer questions) there is some variation between instances of a given item generator, and 2) there are relationships between item generators (e.g., practice on a programming question relating to loops would likely improve students ability to complete a short answer question related to loops and vice-versa).

Nevertheless, for our purposes, we believe this approach is viable. The items of each item generator were considered sufficiently similar by the instructor to be fungible with respect to the exams. Furthermore, the method is robust to whether or not learning occurs *between* subsequent attempts on the same problem or from students attempting a problem, trying new problems, and returning again to an older problem. If the student learns significantly by completing many other homework problems between two attempts at a given problem during the homework period, we can still correctly attribute the learning to having taken place during the homework period. As such, we made no attempt at topic modeling in this work.

4.1 Correctness tracing: post hoc modeling for student knowledge

In general, knowledge tracing (KT) techniques were developed as *predictors* of student performance or *estimators* of the latent knowledge state of students. KT is used either to estimate a student's likelihood of getting the next attempt correct based on previous attempts, adjusting after each success and failure as the student engages with an assessment, or to track changes in students' latent knowledge over time. Much of the difficulty of KT techniques results from attempting to instantaneously obtain a signal of student knowledge as students are engaging with learning opportunities. In our case, we already have all the data from the course as the course has ended and do not need an instantaneous, updating measure of student knowledge. Instead, we desire to perform a *post hoc* analysis of students' submissions to estimate how their learning changed over an entire course's worth of data. Our chosen method, CT, measures students' knowledge as demonstrated by an increase in the likelihood that they would get given items correct more frequently over time.

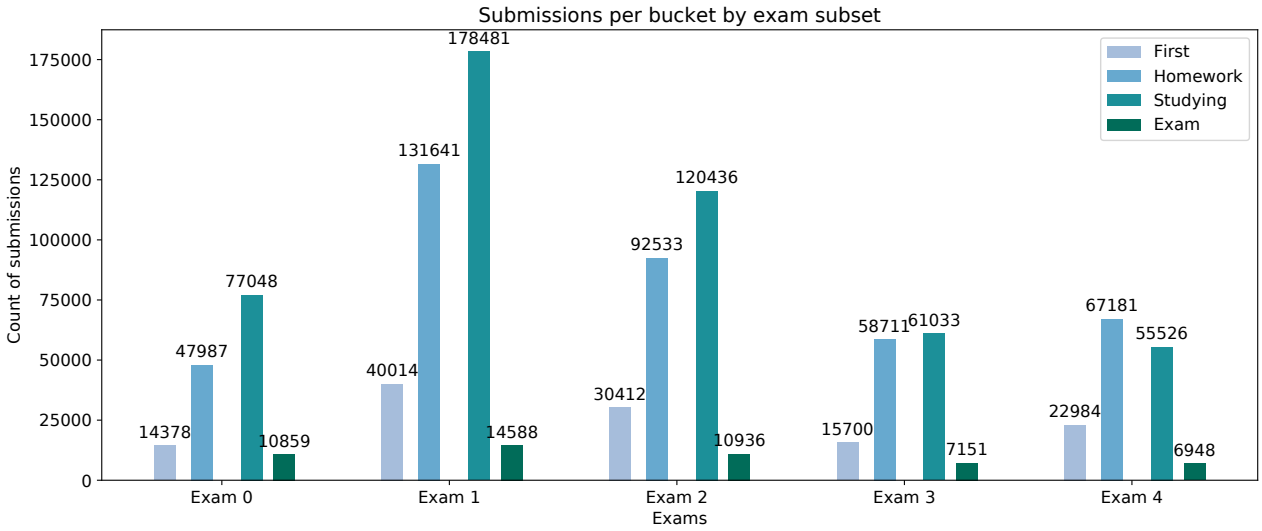

Figure 4: The submission count per period. Total, there are 1,064,547 submissions in our data set. As the semester progressed, homework had fewer but harder problems, which accounts for the reduction in submissions.

The method presented by Chen et al [9] can be summarized by the following formulation:

$$\begin{aligned} &\text{optimize: } \mathcal{L}(p_1, \ldots, p_n; x_1, \ldots, x_n) \\ &\text{subject to: } 0 \leq p_i \leq 1 \text{ for all } i \\ &\qquad p_i \leq p_j \text{ for all } i < j \end{aligned} \tag{1}$$

where $x_1, \ldots, x_n$ is the result of a series of submissions which are either 1 (correct) or 0 (incorrect), and the method tries to find a series of predictions $p_1, \ldots, p_n$ that optimizes the loss function, under the constraints that: (1) $p_1, \ldots, p_n$ are between 0 and 1, as they represent an estimate of the instantaneous probability that the student would get each attempt correct and (2) $p_1, \ldots, p_n$ are monotonically non-decreasing, which is based on the assumptions that the attempts are made over a short enough time period that forgetting is insignificant and additional practice would not hurt a student's ability to answer these questions. Since the homework, practice, and exam attempts occurred over a three-week window, during which there were a lot of related practice, we believe these assumptions are reasonable. Rather than having a model with explicit parameters as found in BKT, the method calculates the probabilities $p_1, \ldots, p_n$ by optimizing them directly for the target loss function. Chen et al have shown that minimizing root-mean-square error (RMSE) and maximizing log-likelihood would yield the same optimal solution under constraints specified in Equation 1.

We chose to use CT over BKT or DKT as it nicely fit our use case. The CT method is able to finely locate and predict the "jumps" in a students' likelihood of getting a question correct when analyzing the data in a *post hoc* fashion, which may be too precise a transition for usual predictive knowledge tracing. For our purposes, a high accuracy, *post hoc* model was ideal for analyzing changing student knowledge as a historical trend from our course's data.

One important weakness of CT, however, is that it is prone to underestimate student knowledge on an incorrect first attempt because the optimizer sets the probability of correctness to be zero so as to minimize error on that attempt. Similarly, the probability on a correct final attempt will always be estimated as 1.0, which may be an overestimate. This potentially could be remedied by adding additional constraints to the method (e.g., limiting the rate of increase), but we did not attempt such constraints with this work.

4.2 Demonstrating CT using "Harlow", a sample student

To clarify our use of CT, we present a walk-through of how the method models our data for one individual and two questions from Exam 3, selected randomly from students whose behavior allows for representative variety in CT's estimates. We refer to the student as Harlow, which is a name that was not present in the actual class. On Exam 3, two of the questions that were randomly selected for Harlow to complete were the programming question `progLargestLessThanValue` and the short answer question `valueOfListReordering`.

Harlow had notably different experiences with these questions; Figure 5 plots the correctness of Harlow's individual submissions as dots that are color coded based on the period in which the submission occurred. With `progLargestLessThanValue`, Harlow made two attempts on homework to get the question correct once, got it correct once on a practice exam with a single attempt, and tried it twice on Exam 3 without getting a correct answer. With `valueOfListReordering`, Harlow had 9 attempts on homework with 6 correct submissions, 4 encounters across two practice exams for 2 correct submissions total, and a correct answer as the only attempt on Exam 3.

Figure 5 also shows the result of running CT as a line indicating the instantaneous estimate of Harlow's likelihood of getting the question correct. In both cases, Harlow got the first attempt wrong, so the model assign's Harlow's likelihood of getting the question correct as 0%, so as to minimize the error relative to the actual outcome. While Harlow is flipping between correct and incorrect attempts, the model computes a likelihood of correctness for each attempt

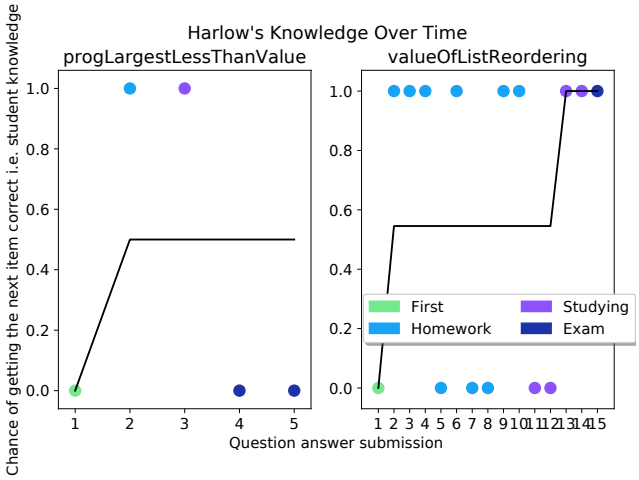

Figure 5: The results of running CT on Harlow's answers to our two selected questions. Harlow eventually appeared to learn how to do *valueOfListReordering*. However, Harlow's ability to complete the programming question never stabilized, so the model never attributed more than a 50% chance that Harlow had learned the question's material.

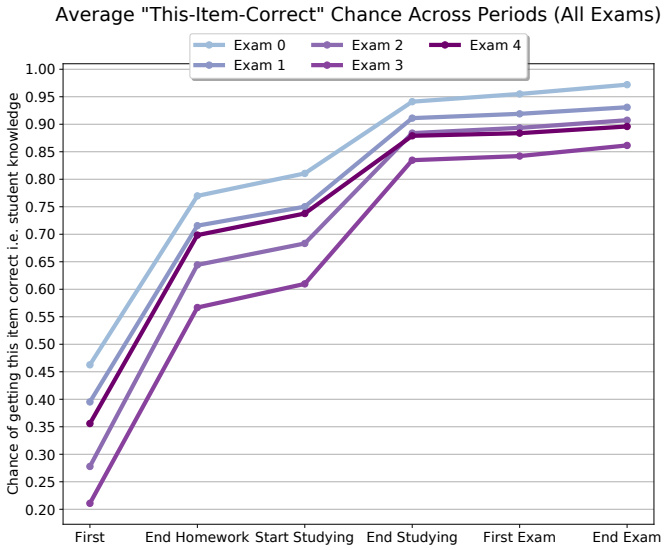

Figure 6: The changing average "this-item-correct" chance from CT per period. CT suggests the majority of student learning is occurring during the homework period, although the study period is also significant.

that minimizes the error for those correct and incorrect attempts, constrained to be non-decreasing. Because Harlow's last three attempts at `valueOfListReordering` were all correct, the model decides that Harlow has mastered the question with a 100% likelihood of getting the question correct.

We ran CT for each student on each question independently. From each trace, we extract six estimates of the student's likelihood of getting a question right: their first and last attempts in the homework period (*First*, *End Homework*), their first and last attempts in the study period (*Start Studying*, *End Studying*), and their first and last attempts on the exam (*First Exam*, *End Exam*). Any student without a submission in that period (i.e., students who did not study or students who did not get that question on their exam) has their previous submission to that point in the timeline used in compliance with CT's assumption that students do not forget. We then average these likelihoods across all students and all questions for a given exam period. This allows us to explore the changing student knowledge as an average for all the students in a course across the different learning opportunities presented by homework, studying, and assessment.

5. RESULTS

5.1 CT attributes significant learning to both the homework and study period; homework contributes slightly more

The results of running CT are shown in Figure 6. From the slopes of the lines, it can be seen that CT estimates that more learning is occurring (i.e., the change in student likelihood of correct attempts is larger) during the homework period than the study period. The plot suggests that the course material tends to get more difficult as the semester progresses, with the initial and final likelihood of correctness both decreasing as we move from Exam 0 to 3. Furthermore, the lines for Exams 0 through 3 show almost identical trends.

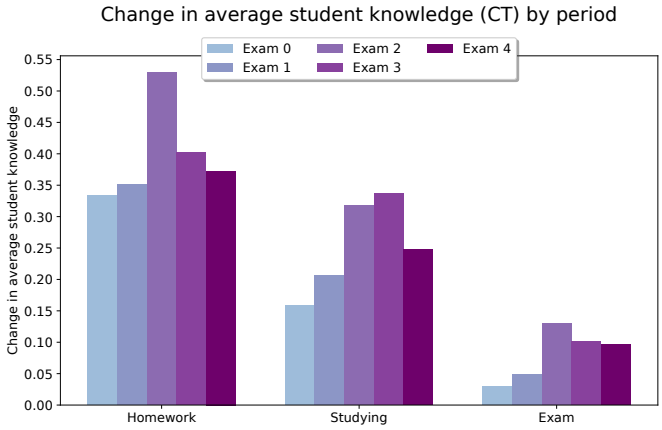

Figure 7: The average change in student knowledge by period according to CT. The largest change occurs during the homework period, with a smaller change from study, and the smallest on exams.

Exam 4, the final exam, behaves differently from the other four exams, which we'll consider in the discussion section.

Figure 7 plots the change in likelihood of correctness from the beginning to the end of each period. When we compare the pre-exam increase in student knowledge (as measured by likelihood of correctness) between the homework and study period, CT attributes 57–65% of the learning to the homework period and 35–43% to the study period, across the five exams. The CT method also attributes some learning to the exam period, which we'll consider in the discussion section.

5.2 Learning trends are largely independent of question type

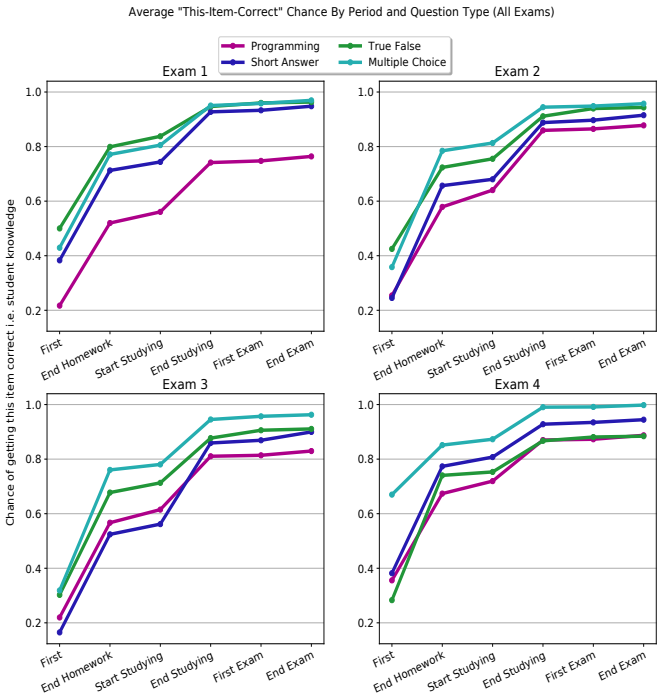

Figure 8: The changing average "this-item-correct" chance from CT for each question type from Exams 1 to 4. Different question types start out with a higher assumed learning to start, which suggest more students got those questions right on their first attempt.

To address **RQ2**, we disaggregated the exam data sets by question type to see whether there was any notable difference between types. For this analysis, we omitted Exam 0, as Exam 0 did not feature programming questions.

Figure 8 shows the per-question type CT results. The only notable finding is that different questions start at different levels of initial student knowledge and end with different amounts of knowledge, which changes the starting and ending points in Figure 8. Because of this, different questions drop off faster than others in terms of how much is learned during the practice period. Generally, students have less to learn with true/false and multiple-choice questions through the practice period than they do on programming and short answer questions, although all question types experience a learning drop-off through to the exam.

6. DISCUSSION AND LIMITATIONS

6.1 RQ1: Students in this course learn slightly more during the homework period than the study period

CT attributes more learning to the mandatory homework period in this particular course. This is represented as the largest increase in student knowledge from their first homework submission to the last. That gives us some confidence that a course with significant homework opportunities does provide students with productive chances to learn as opposed to just inundating students with "busy work."

Interestingly, CT also indicates performance on the exam is better than at the end of the study period. There are a few possible explanations for this. The most likely explanation is that, given the higher stakes of the exam, students are trying harder, resulting in a higher correct rate that is being observed by the model. In addition, some of the score improvements observed on the exam could be attributed to the last pre-exam practice attempt if, for example, the student got the question wrong, but learned from seeing the correct answer. This also might be just be an artifact of CT, as any students that have incorrect and correct attempts to a given question on the exam will have learning attributed to them. Finally, actual learning might be occurring during the exam. The amount of "learning" attributed to the exam period is, however, fairly negligible.

Importantly, one should not attempt to generalize about the learning potential of homework relative to elective practice for all courses from these results. We expect that courses that assign less homework might observe less learning during the homework period and students might compensate by studying more, thereby making more of the learning occur during that optional studying. It could also be the case that there are diminishing returns on each attempt on a specific question, which the first attempt providing the most learning benefit, then the second, decreasing further with each attempt from homework through the study period. It is reassuring, though, to see that this course's homework and study opportunities (i.e., the practice exam generators made available to students) both appear to contribute significantly to student learning.

6.2 RQ2: All question types show similar learning trends

When we disaggregate the analysis by question type, the general shape and progression of results is the same for every question type compared to the source exam. Different questions start with lower amounts of student knowledge, but this appears to mostly be a function of the difficulty of the problem's type: programming and short answer questions, which require more actual coding on the students' parts, tended to start and end lower.

The lack of different behavior when we disaggregate by question type is more interesting than it may initially appear. This means that the "shape" of student learning does not differ significantly with the question type. Given this, it appears that homework and additional studying have the same impact on student results regardless of the kind of question. This does mean there are diminishing returns on easier question types over the period compared to harder ones, but not a deficiency in how homework and practice helps on question types where students still have learning they can do.

6.3 Limitations

There are some obvious limitations to the current work. First, our findings about the relative learning during the homework and study periods cannot be assumed to generalize to other course contexts. Courses with different homework, study materials, and exam structures will likely have different breakdowns of learning in each phase.

Second, CT is a fairly coarse measure of learning. Scores as a performance indicator are not alone proof of student learning. Additionally, CT's potential for underestimating likelihood of correctness of first attempts (by strictly optimizing for RMSE) could make the model overestimate the learning that is occurring in the first few attempts, which is likely occurring in the homework period. We do not have confidence that these measures of learning are particularly precise. While we omit it from the paper, we also ran a regression model to estimate the learning in the same periods of the course. The regression generally showed the same trends as CT, giving us more confidence in CT's results.

Finally, these methods do not disambiguate from learning that happens during the homework and studying periods and learning that occurs specifically from homework and elective practice problems. There are notable reasons to believe that students are learning significantly from reading the textbook, engaging in active learning exercises, and, perhaps, even from listening to the lecturer speak. The learning that occurs during these activities is attributed to the period in which it occurs, rather than to the specific task.

7. CONCLUSION

In this work, we explored the degree to which we can attribute student learning between required homework and elective study performed prior to a summative assessment. To analyze learning, we used a *post hoc* method of "this-item-correct" likelihood (correctness tracing) to estimate student knowledge. We found that (required) homework and (elective) studying both contributed significantly to student learning, with homework contributing slightly more. Further, despite using multiple question types, we found the most notable difference between question types is where student knowledge starts and not the *shape* of their learning improvements.

We think that our results show that frequent, exam-relevant homework and highly-accessible means for study (e.g., practice exam generators) are both effective means of facilitating student learning and believe that these findings could generalize to other contexts. The magnitude of learning from each component may differ, but courses with similar homework and studying opportunities will hopefully see similar learning gains during each period.

There remain areas for future work. Considering data, we only use students' submissions to questions that also appear on homework. Some ability to include other learning events, such as reading a textbook, would give a clearer picture of students' learning process. Additionally, some topic-level labeling might allow us to include questions unique to exams in our data and analysis.

With respect to CT's model, we made no attempt to compensate for the method's tendency to underestimate on initial incorrect attempts. Future work could investigate constraining this behavior by limiting the allowable slope. Further, there is room to adapt the model to using a richer source of information than students' correctness on submissions — for example, by fitting a similar optimization on students' knowledge as estimated by methods such as Item Response Theory (IRT).

8. REFERENCES

[1] E. Bailey, J. Jensen, J. Nelson, H. Wiberg, and J. Bell. Weekly formative exams and creative grading enhance student learning in an introductory biology course. *CBE—Life Sciences Education*, 16(1):ar2, 2017.

[2] J.-A. Baird, D. Andrich, T. N. Hopfenbeck, and G. Stobart. Assessment and learning: fields apart? *Assessment in Education: Principles, Policy & Practice*, 24(3):317–350, July 2017.

[3] R. S. J. d. Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In B. P. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, editors, *Intelligent Tutoring Systems*, pages 406–415, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[4] G. Başol and G. Johanson. Effectiveness of frequent testing over achievement: A meta analysis study. *Journal of Human Sciences*, 6(2):99–121, July 2009.

[5] H. Bembenutty and M. C. White. Academic performance and satisfaction with homework completion among college students. *Learning and Individual Differences*, 24:83–88, Apr. 2013.

[6] J. Bempechat. The motivational benefits of homework: a social-cognitive perspective. *Theory Into Practice*, 43(3):189–196, Aug. 2004.

[7] R. E. Bennett. Formative assessment: a critical review. *Assessment in Education: Principles, Policy & Practice*, 18(1):5–25, Feb. 2011.

[8] P. Black and D. Wiliam. Developing the theory of formative assessment. *Educational Assessment, Evaluation and Accountability(formerly: Journal of Personnel Evaluation in Education)*, 21(1):5, Jan. 2009.

[9] B. Chen, M. West, and C. B. Zilles. Towards a model-free estimate of the limits to student modeling accuracy. In K. E. Boyer and M. Yudelson, editors, *Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018, Buffalo, NY, USA, July 15-18, 2018*. International Educational Data Mining Society (IEDMS), 2018.

[10] P. Chen, Y. Lu, V. W. Zheng, and Y. Pian. Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 39–48, 2018.

[11] S. L. Chew. Helping students to get the most out of studying. *Acknowledgments and Dedication*, page 215, 2014.

[12] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, Dec. 1994.

[13] W. Fakcharoenphol, E. Potter, and T. Stelzer. What students learn when studying physics practice exam problems. *Phys. Rev. ST Phys. Educ. Res.*, 7:010107, May 2011.

[14] B. Gutarts and F. Bains. Does mandatory homework have a positive effect on student achievement for college students studying calculus? *Mathematics and Computer Education*, 44(3):232–244, Fall 2010.

[15] M. K. Hartwig and J. Dunlosky. Study strategies of college students: Are self-testing and scheduling related to achievement? *Psychonomic Bulletin and Review*, 19:126–134, 2012.

[16] S. Irvine and P. Kyllonen. *Item Generation for Test Development*. Lawrence Erlbaum Associates, 2002.

[17] J. A. Johnson and R. McKenzie. The effect on student performance of web-based learning and homework in microeconomics. *Journal of Economics and Economic Education Research*, 14(2):115–125, 2013. Copyright - Copyright Jordan Whitney Enterprises, Inc 2013; Document feature - Tables; ; Last updated - 2020-11-17.

[18] J. D. Karpicke and J. R. Blunt. Retrieval practice produces more learning than elaborative studying with concept mapping. *Science*, 331(6018):772–775, 2011.

[19] M. Khajah, R. V. Lindsey, and M. C. Mozer. How deep is knowledge tracing? *CoRR*, abs/1604.02416, 2016.

[20] M. Khajah, R. Wing, R. Lindsey, and M. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Educational Data Mining 2014*. Citeseer, 2014.

[21] J. Laverty, S. Underwood, R. Matz, L. Posey, J. Carmel, M. Caballero, C. L. Fata-Hartley, D. Ebert-May, S. E. Jardeleza, and M. M. Cooper. Characterizing college science assessments: The three-dimensional learning assessment protocol. *PLoS ONE*, 11(9):e0162333, 2016.

[22] P. Magalhães, D. Ferreira, J. Cunha, and P. Rosário. Online vs traditional homework: A systematic review on the benefits to students' performance. *Computers & Education*, 152:103869, July 2020.

[23] S. Minn, Y. Yu, M. C. Desmarais, F. Zhu, and J. Vie. Deep knowledge tracing and dynamic student classification for knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1182–1187, 2018.

[24] J. W. Morphew, M. Silva, G. Herman, and M. West. Frequent mastery testing with second-chance exams leads to enhanced student learning in undergraduate engineering. *Applied Cognitive Psychology*, 34(1):168–181, 2020.

[25] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In P. De Bra, A. Kobsa, and D. Chin, editors, *User Modeling, Adaptation, and Personalization*, pages 255–266, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[26] Z. A. Pardos and N. T. Heffernan. Kt-idem: Introducing item difficulty to the knowledge tracing model. In J. A. Konstan, R. Conejo, J. L. Marzo, and N. Oliver, editors, *User Modeling, Adaption and Personalization*, pages 243–254, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[27] R. Pelánek. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, 27(3):313–350, Dec. 2017.

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

[29] D. Ramdass and B. J. Zimmerman. Developing self-regulation skills: The important role of homework. *Journal of Advanced Academics*, 22(2):194–218, 2011.

[30] H. L. Roediger and J. F. Nestojko. The relative benefits of studying and testing on long-term retention. *Cognitive modeling in perception and memory: A festschrift for Richard M. Shiffrin*, pages 99–111, 2015.

[31] C. S. Ryan and N. S. Hemmes. Effects of the contingency for homework submission on homework submission and quiz performance in a college course. *Journal of Applied Behavior Analysis*, 38(1):79–88, 2005.

[32] M. L. Still and J. D. Still. Contrasting traditional in-class exams with frequent online testing. *Journal of Teaching and Learning with Technology*, 4(2):30, 2015.

[33] B. W. Tuckman. Using tests as an incentive to motivate procrastinators to study. *The Journal of Experimental Education*, 66(2):141–147, 1998.

[34] C. K. Waugh and N. E. Gronlund. *Assessment of Student Achievement (10th Edition)*. Pearson, 2012.

[35] M. West, G. L. Herman, and C. Zilles. Prairielearn: Mastery-based Online Problem Solving with Adaptive Scoring and Recommendations Driven by Machine Learning. In *2015 ASEE Annual Conference & Exposition*, Seattle, Washington, 2015. ASEE Conferences.

[36] M. West, N. Walters, M. Silva, T. Bretl, and C. Zilles. Integrating diverse learning tools using the prairielearn platform. In *Seventh SPLICE Workshop at SIGCSE 2021 (Virtual event)*, March 2021.

[37] D. Wiliam. What is assessment for learning? *Studies in Educational Evaluation*, 37(1):3–14, 2011. Assessment for Learning.

[38] X. Xiong, S. Zhao, E. G. Van Inwegen, and J. E. Beck. Going deeper with deep knowledge tracing. *International Educational Data Mining Society*, 2016.

[39] L. Zhang, X. Xiong, S. Zhao, A. Botelho, and N. T. Heffernan. Incorporating rich features into deep knowledge tracing. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, page 169–172, New York, NY, USA, 2017. Association for Computing Machinery.

[40] C. Zilles, R. T. Deloatch, J. Bailey, B. B. Khattar, W. Fagen, C. Heeren, D. Mussulman, and M. West. Computerized testing: A vision and initial experiences. In *American Society for Engineering Education (ASEE) Annual Conference*, 2015.

[41] C. Zilles, M. West, G. Herman, and T. Bretl. Every university should have a computer-based testing facility. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU)*, May 2019.

[42] C. Zilles, M. West, D. Mussulman, and T. Bretl. Making testing less trying: Lessons learned from operating a Computer-Based Testing Facility. In *2018 IEEE Frontiers in Education (FIE) Conference*, San Jose, California, 2018.