Understanding the Doer Effect for Computational Subjects with MOOCs

by

Jitesh Maiyuran

B.S., Massachusetts Institute of Technology (2018)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author .....
Department of Electrical Engineering and Computer Science
May 25, 2018

Certified by .....
Una-May O'Reilly
Principal Research Scientist
Thesis Supervisor

Certified by .....
Erik Hemberg
Research Scientist
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chairman, Master of Engineering Thesis Committee

Understanding the Doer Effect for Computational Subjects with MOOCs

by

Jitesh Maiyuran

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2018, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this thesis, we examined the relationship between the doer effect and learning computational subjects. Computational thinking is becoming increasingly important for students and professionals, and teaching this thought process is a relatively new practice. The doer effect is a well-studied learning phenomenon, yet its impact in computational subjects is not well-understood. Also, given that MOOCs cater to a variety of students, predicting student experience levels can benefit instructors. To address these problems, we used data from massive open online courses (MOOCs) to understand how different student activities are correlated with positive learning outcomes. We also considered the doer effect in a variety of scenarios such as prior experience, duration, and course content. Using a variety of linear models and feature engineering methods in the MOOC setting, we were able to replicate the results seen in literature and draw conclusions about the doer effect in new contexts. Because we found prior experience to correlate with student behavior, we also developed a classifier to predict student experience levels given demographic and behavioral data; our model gives strong accuracy and is robust for use in small data sets.

Thesis Supervisor: Una-May O'Reilly
Title: Principal Research Scientist

Thesis Supervisor: Erik Hemberg
Title: Research Scientist

Acknowledgments

I would like to thank Erik Hemberg and Una-May O'Reilly for giving me the opportunity to work on this research project. Their guidance and mentorship allowed me to become a better student and researcher throughout this process; without their help, none of this would be possible. Ana Bell was also incredibly helpful; her intuition for the courses drove our research in the right direction. The entire ALFA group was also very supportive, and I am grateful for the time I spent with them.

Finally, it should go without saying that I would also like to thank my mother, father, and sister for their perpetual support. I am here today because of them.

Contents

1 Introduction **19**

1.1 The Doer Effect 20

1.2 MOOCs and Learning Context 20

1.3 Student Prior Experience 23

1.4 Research Questions and Contributions 25

1.5 Overview 26

2 Exploratory Analysis **27**

2.1 Videos 27

2.1.1 Videos and Grades 27

2.1.2 Videos Over Time 28

2.2 Finger Exercises 30

3 Identifying the Doer Effect Using Existing Methods **33**

3.1 Method 33

3.2 Experiments and Results 35

3.2.1 The Doer Effect in Computational MOOCs 35

3.2.2 The Doer Effect by Prior Experience 37

3.2.3 The Doer Effect in Advanced Courses 38

3.2.4 The Doer Effect in the Long Term 39

3.3 Discussion 40

4 Integrating Enriched Features **41**

4.1 Method . . . 41
4.1.1 Feature Engineering . . . 42
4.1.2 Regression . . . 43
4.2 Experiments and Results . . . 44
4.2.1 The Doer Effect in Computational MOOCs . . . 44
4.2.2 The Doer Effect by Prior Experience . . . 46
4.2.3 The Doer Effect in Advanced Courses . . . 48
4.2.4 The Doer Effect in the Long Term . . . 48
4.2.5 The Doer Effect for Specific Topics . . . 49
4.3 Discussion . . . 51

5 Predicting Prior Experience in a MOOC 53
5.1 Method . . . 53
5.1.1 Hierarchical Models . . . 54
5.1.2 Data . . . 55
5.1.3 Models . . . 56
5.2 Experiments and Results . . . 57
5.2.1 Other pooling methods . . . 62
5.2.2 Shrinkage . . . 62
5.2.3 Accuracy . . . 63
5.2.4 Convergence . . . 65
5.3 Discussion . . . 65

6 Conclusions & Future Work 67

List of Figures

1-1 The edX platform offers students a variety of methods to learn material, Here, we have both a video (left) and two types of optional "finger exercises" that students complete immediately after watching the video. Finger exercises can be coding questions (center) or simpler multiple choice questions (right). 22

1-2 Visualizing the proportions of certified survey respondents allows us to better understand the nature of the courses. We see that in 6.00.1x, most students who received a certificate have some programming experience, and in 6.00.2x, most have taken 6.00.2x with almost no students starting with no experience. 24

2-1 We can observe how videos are related to student performance from two different perspectives: the number of videos watched (left) and the number of actions (pause, play, seek, etc) taken throughout the 6.00.1x course. For each student who received certification in the course, we plotted both of these statistics against the student's final grade. . . . 28

2-2 To understand how students use videos to learn material, we can observe how many times students interact with video content i.e. play, pause and seek specific times in videos on each day of a course. Problem set start and due dates are shown in dotted lines. Some dates overlap i.e. the fifth and sixth problem sets were both due on 10/27. . 29

2-3 Both revealing solutions and checking solutions are indicators of 'doing', but these actions represent slightly different intentions. On the y-axis, 'pc' shows the number of times a student checks a problem and on the x-axis, 'sa' shows the number of times a student revealed the solutions, with marginal distributions as well. 31

2-4 Plotting chapter grades against the number of attempts a finger exercise for all certified students for two chapters, we see different distributions for chapters. Here, each color denotes a specific finger exercise in the unit. We see that the basic programming constructs in Week 1 require fewer attempts in general, while more conceptually advanced topics such as complexity, require more attempts in general. 32

3-1 6.00.1x variation heatmaps where the within-unit quintile is shown on the y-axis and the outside-unit quintile is shown on the x-axis, with the number of students at the intersection shown in each square. In general, we see a concentration of students along the diagonal, indicating that most students fall into the same quintile for both their within-unit and outside-unit activity, though there is enough variation to justify our method. 36

3-2 6.00.2x variation heatmaps, similar to Figure 3-1. The variation lies less along the diagonal we see in 6.00.1x, perhaps due to the more advanced course drawing a wider range of students who are capable of going forward or need to review prior concepts more frequently. Only two students had no coding background, causing highly discrete values for that group. 36

4-1 When looking at a calendar of the course's due dates and when a student takes certain actions, we can compute the corresponding features. The first row of the figure enumerates the days, while the second and third rows show the timespans for two problem sets with the first due on the 12th day and the second problem set due on the 19th day. Each of the eight boxes in fourth row then indicates a certain action taken by the student, with the corresponding unit noted in the box. For example, the first box on the left indicates a finger exercise or video for unit 1. Because the first two actions do not fall in the seven days prior to a due date, they are not counted. Of the three actions completed during the critical time for PS1, two are relevant to PS1, giving the 'within' designation, while the other is for PS2, giving the 'before' designation because the actions occurs before the unit has been covered. A similar classification occurs for the actions taken in days 13-19 before the PS2 due date. 43

5-1 This figure defines one hierarchical model that we evaluated where we consider two variables in our logistic regression for which we have two coefficients β_1 and β_2 and an intercept α. In this model, we assume that these coefficients are independently drawn from normal distributions $N(\mu_1, \sigma_1^2)$, $N(\mu_2, \sigma_2^2)$, and $N(\mu_\alpha, \sigma_\alpha^2)$. Regression coefficients/intercepts are drawn once for each class. Here, we show partial pooling over the possible economic statuses ("developing" or "developed") of the countries of the user. The veteran status of each user is therefore a Bernoulli random variable with $p =$ the inverse of the logit of $X_1\beta_1 + X_2\beta_2 + \alpha$. 55

5-2 A unit test trace of simulated data where two different values of β_1 were used to generate data. The simulated data indicates that the model works as intended with the posterior distributions of β_1 falling on the two simulated values. We would create two classifiers, one where β_1 is equal to the mode of each of the traces. 59

5-3 The trace plots represent posterior distributions for the logistic regression coefficients of a model that is partially pooled on economic group. The flatter distributions correspond to developing economic groups. β_1 corresponds to a student's level of formal education and β_2 corresponds to a student's year of birth. We see two posterior distributions because we've created a two dimensional Gaussian distribution over each coefficient to allow for an estimate over each class of data. The traces are noisier beacause they must fit the data in each class while still being constrained by the priors on the parameters. 60

5-4 The trace plots represent posterior distributions for the logistic regression coefficients of a model with unpooled data i.e. each economic group treated independently. The flatter distributions correspond to developing economic groups. β_1 corresponds to a student's level of formal education and β_2 corresponds to a student's year of birth. While this is similar to the trace plots for the partially-pooled model, we see smoother estimates of the posterior because we are no longer constraining the coefficients for each class to be drawn from the same distribution. 61

5-5 The x-axis is β_1 and the y-axis is β_2. The red scatter plot represents the model parameters over the two clusters in the economic groups when unpooled. The blue plots represent the economic groups when pooled. The arrows show the movement in the coefficient space, showing how the parameters exhibit some shrinkage when partial pooling is implemented. 64

5-6 The x-axis is β_1 and the y-axis is β_2. The red scatter plot indicate the coefficients for each region with full pooling, and the blue scatter plot indicates the coefficients when partial-pooling is used. Here we see that partial pooling allows for the regions to take on very different parameters. Regions with less data cling to the mean 64

List of Tables

1.1 6.00.1x and 6.00.2x graded activities over a ten-week course 21

1.2 The size of the courses in terms of available material, with 6.00.1x having many more videos and finger exercises available to students. . 21

1.3 The three main facets of the edX courses that we will consider are videos, 'problem check' (checking an entered solution) and 'show answer' (revealing a solution). 23

1.4 The prior survey responses for 6.00.1x (left) and 6.00.2x (right) highlight the makeup of experience levels of students who enroll in the course. In both courses, we see that inexperienced students enroll in the course, yet relatively few of them complete the course to receive certification. We also see a high dropout rate, typical in MOOCs [16]. 24

3.1 All actions taken preceding the unit due date are culled and binned according to these definitions, analogous to the definitions in prior works. 34

3.2 Regression coefficients and p-values where problem set performance is a function of watching videos and completing problems over different time scales. We have two models here: one for all students ('All' on the far left), and another where students are differentiated by their prior experience level. We only show coefficients significant at the p=0.05 level. 38

3.3 Regression coefficients and p-values obtained using the same method for 6.00.2x. Our findings from the 6.00.2x data are much noisier, yet we still see the within-unit finger exercises being the strongest indicator of success. 38

3.4 In the long term, we also see that finger exercises are more correlated with problem set success with the effect being three times that of videos. 39

4.1 This regression models problem set grades as a function of student activity for 6.00.1x problem sets. Each row is a different affordance/timeframe for which a user can complete activities for a unit. The columns indicate different subsets of students based on their prior experience indicated in Table 1.3, with the first column describing all students. An activity is considered relevant if it is completed in the seven days preceding the problem set due date. Only coefficients significant at the p=0.05 level are included. 45

4.2 We measure the strength of certain actions for certain groups of prior experience groups by expressing $R_s^f = \frac{\beta_s^f}{\beta_{all}^f}$ for each prior experience group and for each of the within-unit actions. Higher ratios indicate that this action is more highly corrrelated with success compared to the general student cohort. We only list coefficients where they are statistically significant. Note that watching videos is far stronger for veterans compared to the general population and checking solutions is not even relevant for veterans but strong for all other groups. 45

4.3 To offer an alternate method of comparing the strength of doing among the student groups, we used a linear model similar to that in Chapter 3; we have 45 coefficients, nine for the features across the five survey groups. We also include mixed effects for unit and user. We see that it is more difficult determine differences among the covariate coefficients due to similarities in their magnitudes and the overlap in the 95% confidence intervals (not shown). 46

4.4 These regressions utilize the same model as those in Table 4.1, though for 6.00.2x. The corresponding groups are for 6.00.2x, for which the survey responses were slightly different. Note that while a 'no experience' option existed in the survey, only 2 of these students completed the course, so a meaningful regression was not possible. Again, only coefficients significant at the $p = 0.05$ level are included. 47

4.5 This regression models problem set grades as a function of student activity for the 6.001x final exam. On the left are the different (learning mode, time scale) combinations for which students can complete activities for a particular unit. The columns indicate different subsets of students based on their prior experience, with the first column describing all students. Because we are modeling the final exam, there is no distinction for "within-unit" activities, and the student's activity over the entirety of the course is used to construct features. Only coefficients significant at the p=0.05 level are included. 49

4.6 To better study specific subjects, we formulate our regression problem to predict student grades on problems in a problem set that assess a certain topic; the features, are constructed similarly as for the previous results, but here, we only select videos that correspond to the topic as well as specific finger exercises that also correspond to the topic. Thus while the content for a particular unit may include many topics, we seek to isolate specific topics. All coefficients listed are significant at the p=0.01 level 50

5.1 We use a mix of demographic and behavioral features to classify students. The 'UN Economic Group' tells us whether the user's country is classified by the UN as 'developed' or 'developing'. Likewise, the 'UN Major Region' tells us the region such as 'Western Europe,' 'Southern Asia,' etc, which is slightly less granular than the user's actual country or even city. Additionally, we map education levels to the equivalent years of schooling i.e. five for elementary school, twelve for high school, sixteen for undergraduate. Behavioral information is collected from the entirety of the student's enrollment in the course; features generally count the number of times a student took an action on the edX platform such as pausing a video or revealing a solution. 56

5.2 We can understand our regression coefficients by considering $\exp \beta$, which gives us the odds ratio if a variable increases by 1. Observing some of the values, we see that having more events i.e. clicks, substantially increases the odds that a user is a veteran as does seeking out specific points in the course videos. We also see that higher years of birth decrease odds of a veteran, which is also sensible (higher birth year = younger). 58

5.3 While our test set is balanced, the number of veterans in developing countries is far fewer than that in developed countries. 63

5.4 We indicate predictive accuracies on a balanced test set of size $n = 14$. We see that the accuracy is similar across all models with a model Partially Pooled on Major Region giving the highest accuracy. 64

Chapter 1

Introduction

Massive Online Open Courses (MOOCs) have gained a large audience in recent years [2, 6]. With greater access to the Internet for users as well as content providers, online courses cover many knowledge domains and cater to large populations. While MOOCs offer many different learning resources such as videos, readings, and practice exercises, understanding exactly how students best learn new material is still unknown. Prior research has shown strong results for the presence of a "doer effect" – the idea that "doing" (i.e. completing practice problems and exercises) contributes significantly to a learner's understanding of the material beyond just watching videos or reading articles [13]. In particular, the literature suggests that learning by doing is more highly correlated with success on assessments of the same material compared to reading or watching videos.

When studying the doer effect in MOOCs, however, there are many nuances to consider given the more open and technology-driven nature of the educational setting. Traditionally, videos have been the primary offering of MOOCs due to the ease of establishing videos online. Lecture notes and accompanying readings are also common due to the ease of incorporation. In contrast, offering online exercises that many online learners can complete is much more challenging for course providers. Effective platforms for such exercises often offer additional features such as hints to guide students as well as problem solutions, e.g. personified programming feedback improves novice programmers' learning and can be used to predict abandonment [14, 22]. Cre-

ating this environment is doubly difficult for computational courses because test cases must be written to evaluate the correctness of solution, which is time-consuming for any instructor. Because of these additional needs, learning by doing is less commonly available to students online. Thus, understanding whether and how learning by doing compares to traditional methods can help us decompose the learning process in order to better tailor general purpose educational materials to the needs of students.

1.1 The Doer Effect

We take a special interest in the doer effect because of the canonical belief that learning to program requires learners to actually program as opposed to simply read about programming. Because of this belief, we also seek to make clear the definition of *doing*: Doing is the completion of exercises that requires the student to answer a question outside the context of an assessment. It is also important to distinguish this from *active learning*, which may just be classified as learners rereading written materials or rewatching videos, which can imply a higher comprehension of the presented material.

1.2 MOOCs and Learning Context

Here we focus on courses that teach 'computational thinking', a term in education to describe a range of curriculum from math to programming to algorithm design. We use the definition developed by Wing 2010 which considers computational thinking to be 'the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine, or more generally, by combinations of humans and machines' [1]. This definition has motivated other computational thinking learning initiatives as well, making it a proper definition to consider in the MOOC setting as well [21].

Massive Open Online Courses (MOOCs) are the second characteristic of our setting, and have their own idiosyncrasies when compared to a traditional classroom

6.001x		6.002x	
Assignment	Due Date	Assignment	Due Date
Python Basics	Week 4	Optimization	Week 5
Simple Programs	Week 5	Randomness	Week 6
Midterm Exam	Week 5	Midterm Exam	Week 7
Structured Types	Week 7	Statistics	Week 8
Good Programming Practices	Week 8	Modeling and Fit	Week 9
Object Oriented Programming	Week 9	Final Exam	Week 10
Algorithmic Complexity	Week 9		
Final Exam	Week 10		

Table 1.1: 6.00.1x and 6.00.2x graded activities over a ten-week course

Course	# Videos	# Finger Exercises
6.00.1x	81	555
6.00.2x	43	177

Table 1.2: The size of the courses in terms of available material, with 6.00.1x having many more videos and finger exercises available to students.

setting. MOOCs often have low completion rates (9-10%), and the lack of certification can deter many students who enroll for professional development. Some MOOCs allow students access to materials and encourage them to learn at their own pace while others enforce a schedule. Other qualities like proctoring, prior experience, and automatic grading can also change the MOOC experience [16]. In this study, we will be considering the edX MOOC platform, where many of these qualities will become relevant.

In order to study the presence of the doer effect in computational learning, we examine the behavior of students in two courses: 6.00.1x, Introduction to Computer Science and Programming Using Python, and 6.00.2x, Introduction to Computational Thinking and Data Science. 6.00.2x covers more advanced concept and is intended to be taken after 6.00.1x. Courses are divided into multiple units, where each unit has an associated problem set, for which students receive grades; a list of assessed material is shown in Table 1.1. We additionally note that the courses differ in the number of materials available to students, seen in Table 1.2. Although edX offers courses that

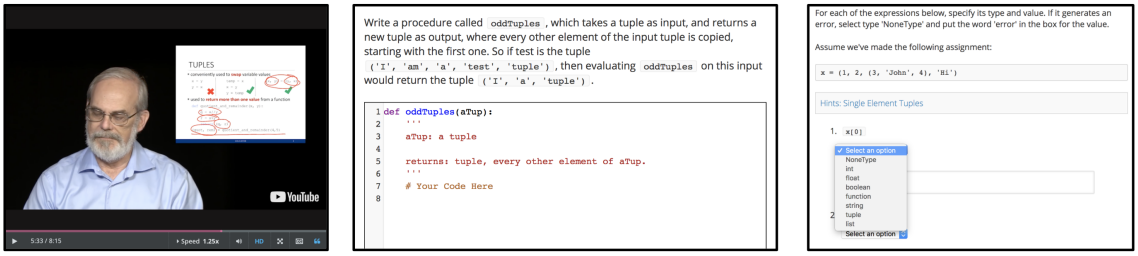

Figure 1-1: The edX platform offers students a variety of methods to learn material, Here, we have both a video (left) and two types of optional "finger exercises" that students complete immediately after watching the video. Finger exercises can be coding questions (center) or simpler multiple choice questions (right).

can be taken at leisure, the courses we examine abide by a schedule; each course is ten weeks long, with problem sets due almost every week after the first few weeks. In order to learn the content for each unit, students are expected to watch lecture videos narrated by instructors and complete "finger exercises" - optional problems interspersed in lecture videos that teach the content discussed in the video. Figure 1-1 shows an example of a finger exercise where students are expected to implement the topics explained in a video. In addition to completing problem sets for grades, students are also assessed via two exams: a midterm and final.

Finger exercises are of special interest to us because they indicate optional "doing" – students completing practice activities on their own accord in order to better understand the material. Furthermore, when completing finger exercises, students can have two options; they can either check their solutions ("problem check") or they can just reveal the solution ("show answer"). A summary of the different affordances offered by edX is shown in Table 1.3. Analyzing how students' interactions with finger exercises relate to their problem set and final exam grades allow us to better understand the doer effect in this setting. We believe that usage of these two functions when completing finger exercises serve as a sufficient proxy for doing.

The edX platform allows students to elect to receive a certificate of completion for completing the course with a passing grade. These are the learners for whom we analyzed data because they would have a proper incentive to learn the material.

Interaction	Description
video	Videos vary in length, and cover topics throughout the course. Students can watch any of the videos at any time as well as drag a slider to select a certain part of the video.
problem check	After entering a solution, all problems (both finger exercises and problem sets) allow students to check their solutions. Students can continue checking solutions regardless of the outcome.
show answer	Students can choose to reveal the solution to problems included in the finger exercises.

Table 1.3: The three main facets of the edX courses that we will consider are videos, 'problem check' (checking an entered solution) and 'show answer' (revealing a solution).

1.3 Student Prior Experience

One aspect of the doer effect that remains unstudied is the effect of prior experience. In particular, do we expect students with equal experience with the subject matter to have grades that are similarly correlated with doing? Studies report different impacts on learning outcomes for introductory computer science courses in traditional classrooms [20, 18]. In order to study the effect of prior experience, we will employ a survey conducted in 6.00.1x and 6.00.2x that asks students to report their familiarity with programming in general. Note that of the many thousands of students who register for the course, a relatively small fraction answer the survey, but because we select only the students who paid to receive certification, many of the students who chose to receive a certificate also answered the survey. This is unsurprising considering that online courses and MOOCs often have low retention rates [10, 15]. The number of students for each survey response is shown in Table 1.3 and Figure 1-2. In 6.00.1x we see that students with no experience are represented as well as students who know a different language when considering the students who enrolled. However, when looking at the students who completed the course, students who knew a different language to begin with outnumber students who had no experience 2 to 1. This disparity gives us a preliminary indication that prior knowledge plays a major role in student success in this course.

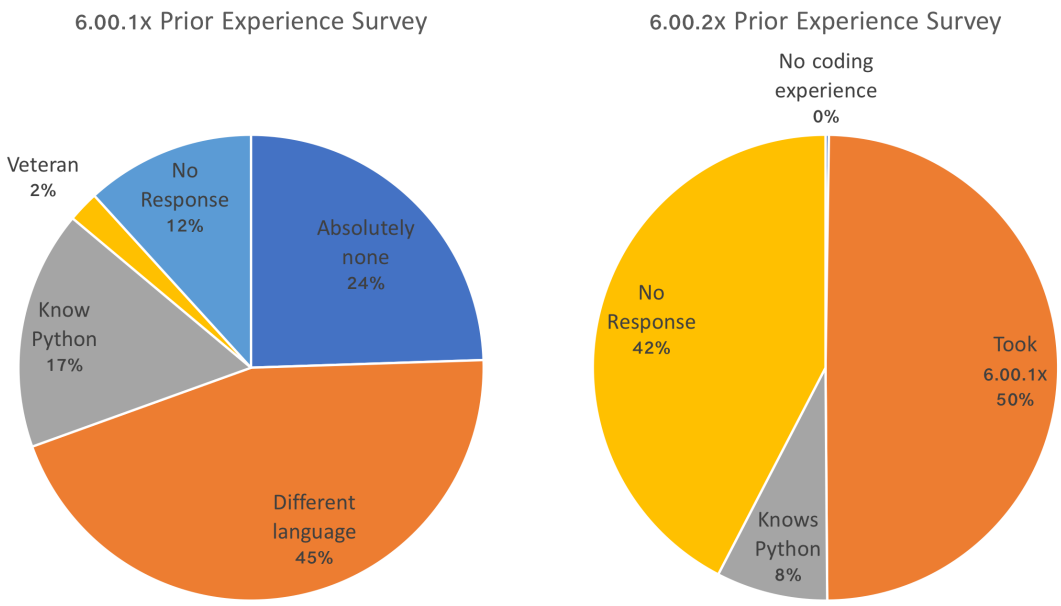

Figure 1-2: Visualizing the proportions of certified survey respondents allows us to better understand the nature of the courses. We see that in 6.00.1x, most students who received a certificate have some programming experience, and in 6.00.2x, most have taken 6.00.2x with almost no students starting with no experience.

Response	Enrolled	Certified
Absolutely none	27,108	853
Different language	27,157	1,569
Know Python	7,934	578
Veteran	1,202	75
No Response	182,552	410
Total	245,953	3,485

Response	Enrolled	Certified
No coding experience	595	2
Took 6.00.1x	3,532	423
Knows Python	1,880	66
No Response	36,666	361
Total	42,673	852

Table 1.4: The prior survey responses for 6.00.1x (left) and 6.00.2x (right) highlight the makeup of experience levels of students who enroll in the course. In both courses, we see that inexperienced students enroll in the course, yet relatively few of them complete the course to receive certification. We also see a high dropout rate, typical in MOOCs [16].

1.4 Research Questions and Contributions

Understanding the role of the doer effect in learning computational subjects is multifaceted, as is our approach. We thus present the following research questions.

We first want to know whether the doer effect exists when learning computational subjects. We then consider altered scenarios: Is the doer effect as prominent for students of different experience levels? Is the doer effect similar for more advanced subjects? Does the doer effect vary for short-term vs long-term learning? Does the doer effect vary for different computational subjects?

To address these questions, we replicated the methods seen in literature [13] and also ran similar experiments with data from different cohorts of students by scraping prior experience levels from the edX platform. Our findings in computational subjects are similar to those presented in the literature. Because MOOCs often offer more information about how a student behaves on the platform, we also implemented a different feature-engineering method that takes advantage of more granular MOOC data, obtaining results that both support and dispute some findings in the literature.

Because we found a strong relationship between doing and student experience, we believe that identifying students with high levels of experience ('veterans') can be valuable for statistical analysis as well as for instructors; this led us to one final question: Can we predict how much experience a student has prior to taking the course? We developed a partial-pooling hierarchical model that determines whether a student is a veteran based on demographic and biographical information.

Having addressed these research questions, it is important that we also integrate the methods into the MOOC-Learner Project (MLP). MLP develops insights from large sets of MOOC data; by integrating our methods to examine the doer effect, instructional designers and learning scientists can extract similar insights from their courses to understand the extent of the doer effect across many domains.

1.5 Overview

In Chapter 2, we will begin by parsing the data to better understand how students behave and interact with the materials available in the course. Having an intuition for the data, we begin to address our research questions. First, in Chapter 3, we adopt the method of previous works, independent of the details of our course and platform [13]. Then, in Chapter 4 we will refine our method to take advantage of the more granular data for each student to highlight the doer effect. Because MOOCs are inherently technology-driven, we can leverage this aspect to develop models more attuned to the MOOC environment, as suggested by prior work [5]. We will see that prior experience can differentiate the extent to which the doer effect is related to student learning. To further address this, in Chapter 5, we will explore methods to classify students by prior experience level in a MOOC setting. Finally, in Chapter 6, we summarize our findings and delineate future directions for our work.

Chapter 2

Exploratory Analysis

Before addressing our research questions, it is informative to probe aspects of our data that provide an intuition for the nature of the courses and student behaviors. In particular, we wish to understand how students interact with the two major learning affordances: videos and finger exercises. In this case, we will be limiting our analysis to students who received a certificate of completion in the course.

2.1 Videos

Videos are one of the most common learning methods, often consisting of recorded lectures that cover course material. On the edX platform, videos and relevant finger exercises are interleaved, allowing students to alternate between an instructor explaining a concept and the student being given an opportunity to exercise that knowledge.

2.1.1 Videos and Grades

To understand the role that videos play in student learning, we can first look at how video-watching is related to student grades, shown in Figure 2-1. When looking at the number of unique videos viewed, we see that some students are not viewing many videos, yet still doing well in the course (upper left region). Meanwhile, most students

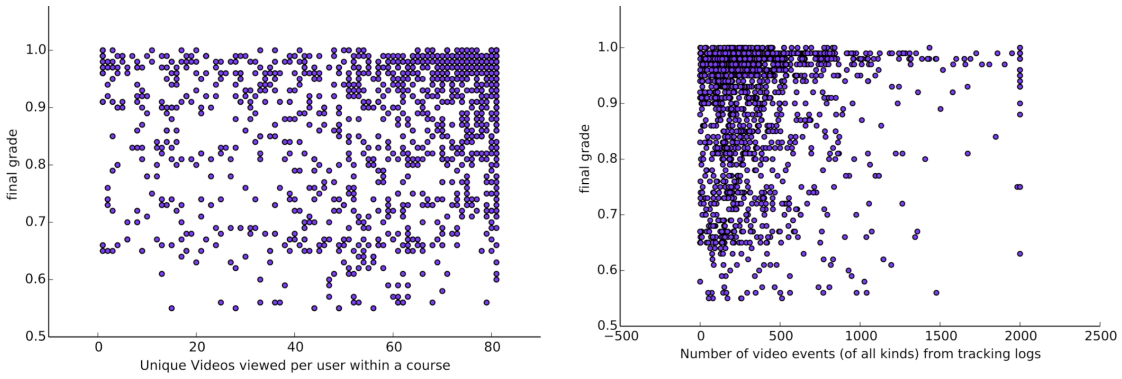

Figure 2-1: We can observe how videos are related to student performance from two different perspectives: the number of videos watched (left) and the number of actions (pause, play, seek, etc) taken throughout the 6.00.1x course. For each student who received certification in the course, we plotted both of these statistics against the student's final grade.

lie in the upper right region, watching all or most videos and still doing well. We see a similar pattern when looking at the number of video events: Most students interact with the videos less than 1000 times, with a few students interacting with the videos much more frequently. Both of these images indicate that video-watching is a major mode of learning offered by MOOCs for learning content.

2.1.2 Videos Over Time

Knowing that videos play a large role in student learning, we can also consider how video-watching is related to students completing individual problem sets. Because instructors often provide the intuition in videos, and this intuition is later evaluated on problem sets, we would expect students to watch videos frequently when completing problem sets. To understand this relationship, we can examine how much video-watching occurs in relation to the calendar of the course – namely, due dates of problem sets and exams, shown in Figure 2-2. We initially see that video activity spikes on days when problem sets are made available to students and when problem sets are due, with the intervening days seeing fewer activity. This tells us that not only do videos seem to correlate with higher final grades, but they are also heavily used when completing problem sets in the short term.

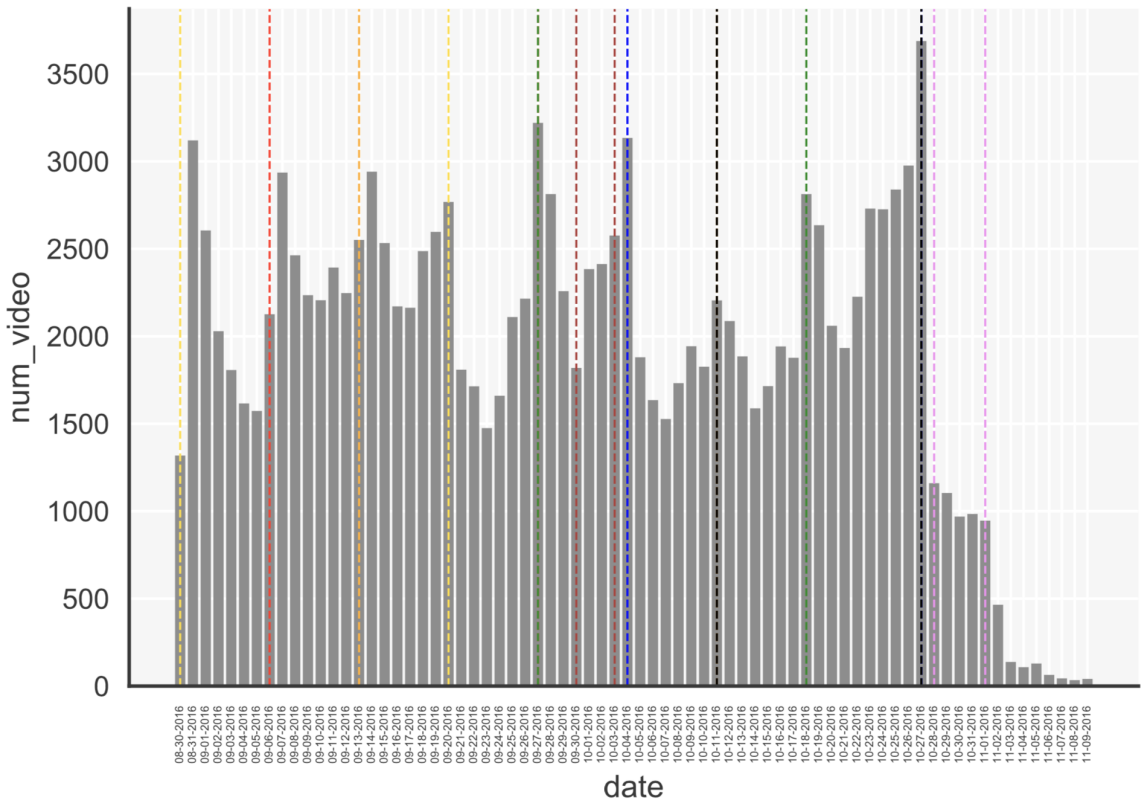

Figure 2-2: To understand how students use videos to learn material, we can observe how many times students interact with video content i.e. play, pause and seek specific times in videos on each day of a course. Problem set start and due dates are shown in dotted lines. Some dates overlap i.e. the fifth and sixth problem sets were both due on 10/27.

2.2 Finger Exercises

The finger exercises described in Chapter 1 are the second essential mode of learning offered to edX students. For each topic/unit, finger exercises offer students an opportunity to implement the idea being discussed. When students struggle, they can elect to reveal the solution or check their solution. We consider finger exercises a proxy for 'doing', and we thus use the number of times a student elects to reveal or check solutions to finger exercises as a measurement of 'doing'. However, we also must understand the differences between these two actions; checking a problem repeatedly is very different from revealing the solution. To understand these differences, we look at the joint distribution over these two actions in Figure 2-3. We see that the number of 'problem check' is significantly greater than the number of 'show answer' for most students, which matches our expectation that students would use the problem-check option multiple times while attempting to solve a problem. This confirms our intuition that while both of these actions are involved with doing they can have very different meanings for a student's learning process.

In addition to examining the doer effect in general, we also wish to examine certain fundamental ideas in computational learning and determine whether the doer effect exists to the same degree in each of these topics or how it differs. Understanding this relationship begins with considering how different finger exercises relate to chapter grades i.e. on which finger exercises is success most correlated with high performance. To explore this idea, we plotted the number of attempts taken on finger exercises against performance on the chapter, shown in Figure 2-4. We see that the relationship between working on an exercise and doing well on the corresponding problem set varies. Considering the basics of Python, we see that students take fewer attempts to get the same grades as they did during Week 6 when learning complexity, a more conceptually advanced topic. This indicates that learning a particular concept is highly dependent on the specific type of practice. By isolating these specific types of practice, we can perhaps get more insight into how students learn these concepts.

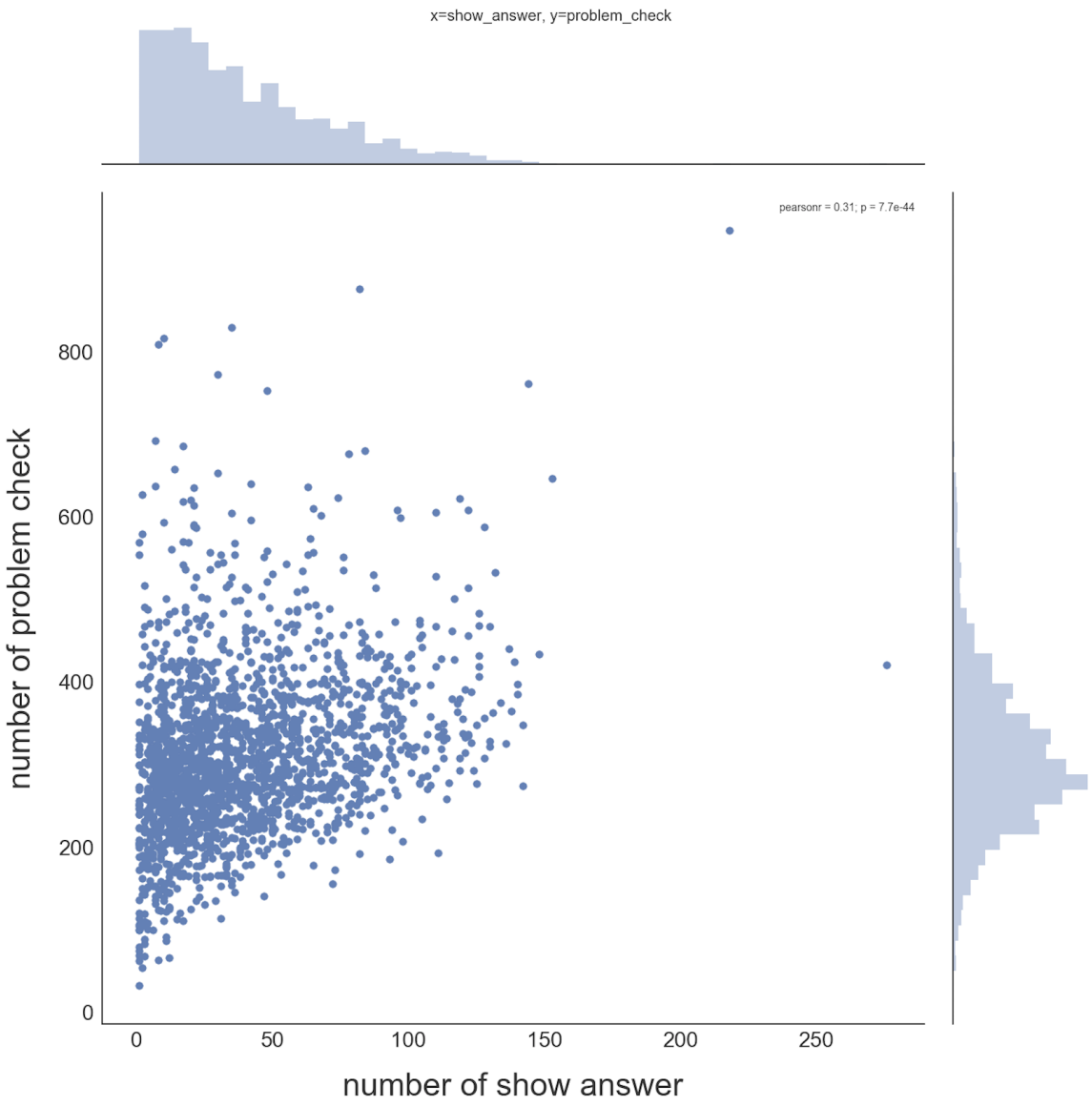

Figure 2-3: Both revealing solutions and checking solutions are indicators of 'doing', but these actions represent slightly different intentions. On the y-axis, 'pc' shows the number of times a student checks a problem and on the x-axis, 'sa' shows the number of times a student revealed the solutions, with marginal distributions as well.

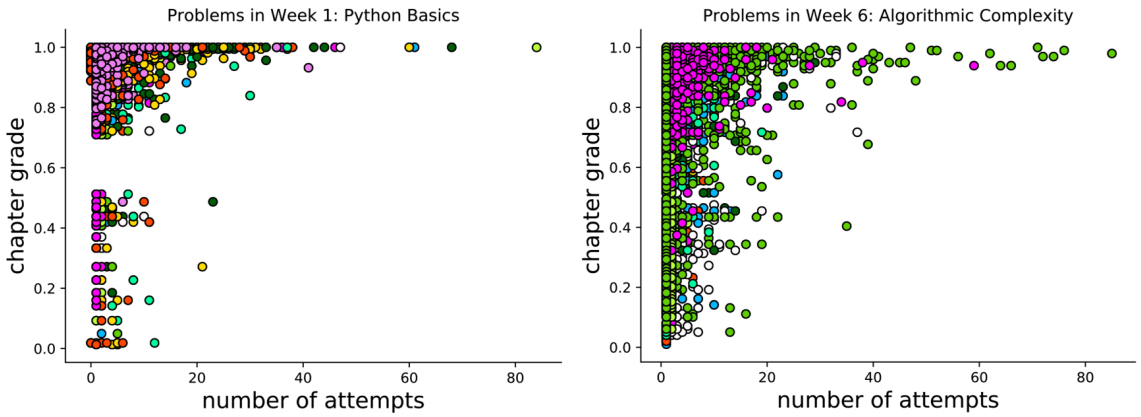

Figure 2-4: Plotting chapter grades against the number of attempts a finger exercise for all certified students for two chapters, we see different distributions for chapters. Here, each color denotes a specific finger exercise in the unit. We see that the basic programming constructs in Week 1 require fewer attempts in general, while more conceptually advanced topics such as complexity, require more attempts in general.

Chapter 3

Identifying the Doer Effect Using Existing Methods

Having a better understanding of how students behave in a MOOC setting, we can now consider how the doer effect relates to student performance. Using the two major modes of learning offered on the edX platform discussed in Chapter 2, videos and finger exercises, we can quantify student behavior and identify the correlation between doing and student performance.

3.1 Method

Using the data discussed in Chapter 1, we will use methods founded on doer effect literature to study how the doer effect manifests in computational learning. [13]. Because students have access to a variety of resources on the edX platform, we consider two main modes of learning: watching videos and completing finger exercises.

When culling actions for a specific unit and student, we consider all actions taken by the student from the beginning of the course until the due date of the unit's problem set. We then quantify these two actions as follows: Each unit in the course has an associated set of video lectures for students. We consider a video 'watched' if the student started watching the video. Similarly, for each unit, there is an associated set of finger exercises. We consider a finger exercise 'done' if the student either

Name	Description	Analog [13]
prereq-unit	A video or finger exercise that is associated with a unit that precedes (i.e. is a prerequsite for) the content being assessed in this problem set.	before
withinreq-unit	A video or finger exercise that is associated with the same content begin assessed in this problem set.	within
postreq-unit	A video or finger exercise that is associated with a unit after (i.e. is a postrequsite for) the content being assessed in this problem set.	after

Table 3.1: All actions taken preceding the unit due date are culled and binned according to these definitions, analogous to the definitions in prior works.

revealed the solution to the exercise or checked their own solution to the problem. We then take both the video and finger exercises completed in this time span and group them by the unit for which they are completed; all of these watched videos and attempted exercises fall into one of three bins: covering material from a prerequisite unit ('prereq-unit'), covering requisite material from the current unit ('withinreq-unit'), and covering material from a future unit ('postreq-unit') i.e. a postrequisite. Note that in our analysis we refer to 'prereq-unit' and 'postreq-unit' as 'outside-unit' to generalize material that is not associated with the unit being assessed. These definitions are analagous to those in Koedinger 2016 as illustrated in Table 3.1 This method constructs six features for each student-unit: three time-scales for videos and three time scales for finger exercises. Note that in this scenario, redoing a finger exercise or rewatching a video is not captured; the value of each unstandardized feature is limited by the number of finger exercises and videos available either for the current unit, for all future units, or for all previous units. We then standardize these values for each unit and use a linear regression model with fixed effects for the six features and random effects for the unit and the student to account for varying difficulties of units and varying student ability.

We also consider an important diagnostic for this method: to give meaningful results, each student must vary their within-unit activity and outside-unit activity

(where 'outside-unit' is simply the sum of 'prereq-unit' and 'postreq-unit'). If every student-unit had the same amount of within-unit and outside-unit activity, we would not be able to identify differences in doing relevant content ('within-unit') vs less relevant content ('outside-unit').

3.2 Experiments and Results

We first consider the presence of variation in student behavior. In particular,we will use the same methods in previous works: considering which quintile a user falls into for both his within-unit and outside-unit activity, shown in Figures 3-1 and 3-2. Looking broadly at the heatmaps for prior experience groups in both 6.00.1x and 6.00.2x, we see a concentration of students along the diagonal where within-unit quintile is equal to outside-unit quintile. This pattern is expected because it indicates that most student-units fall into the same within-unit quintile and outside-unit quintile. However, we also see a great deal of variation, where many student-units are off the diagonal as well, indicating that the within-unit and outside-unit features are still very informative. Because the heatmaps display a great deal of variation over student behvaior with respect to within-unit and outside-unit activity, we can move forward with the mixed-effects regression.

3.2.1 The Doer Effect in Computational MOOCs

We begin our analysis considering all students in 6.00.1x, where finger exercises and video activity on different time scales are fixed effects and the student and unit are random effects, giving the following regression formula in R:

```
lmer(unit_grade.Z ~ num_prereq_fex.Z + num_postreq_fex.Z +
num_withinreq_fex.Z + num_prereq_vid.Z + num_postreq_vid.Z +
num_withinreq_vid.Z + (1|unit) + (1|username), data=edx_data).
```

The results of this analysis are shown in Table 3.2.2, where we examine the coefficients in the first two columns. We see that completing the requisite finger exercises for a unit (0.37) is almost three times as correlated with problem set grades as watch-

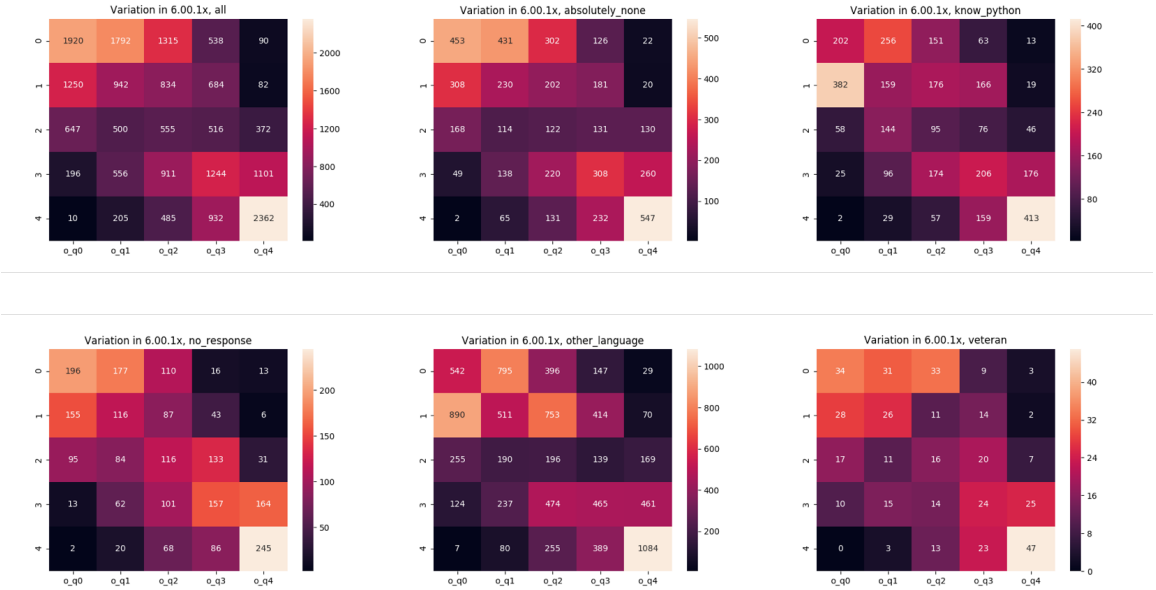

Figure 3-1: 6.00.1x variation heatmaps where the within-unit quintile is shown on the y-axis and the outside-unit quintile is shown on the x-axis, with the number of students at the intersection shown in each square. In general, we see a concentration of students along the diagonal, indicating that most students fall into the same quintile for both their within-unit and outside-unit activity, though there is enough variation to justify our method.

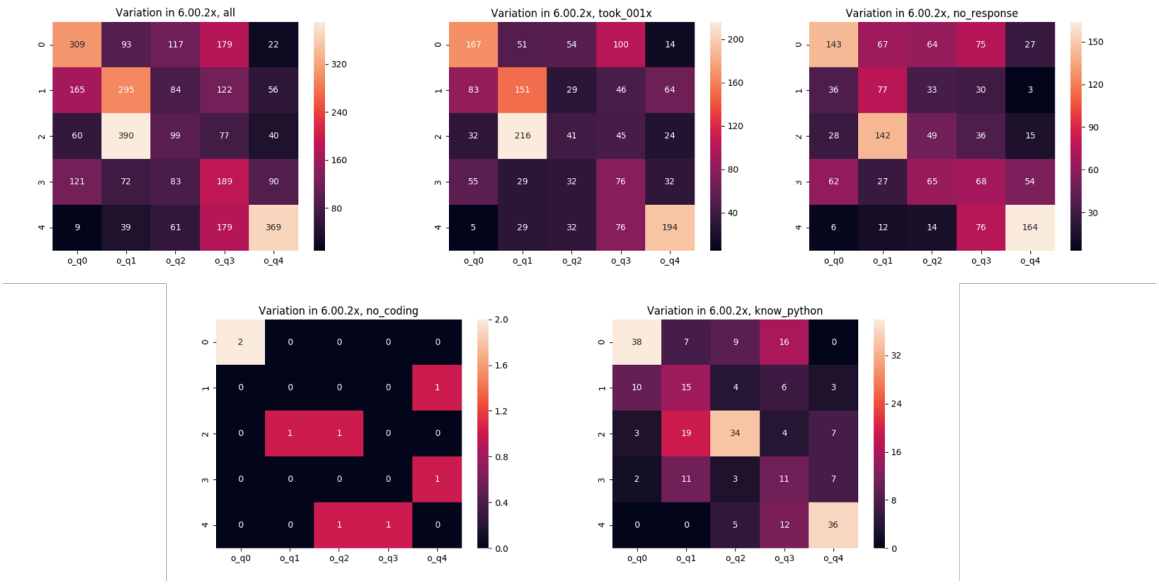

Figure 3-2: 6.00.2x variation heatmaps, similar to Figure 3-1. The variation lies less along the diagonal we see in 6.00.1x, perhaps due to the more advanced course drawing a wider range of students who are capable of going forward or need to review prior concepts more frequently. Only two students had no coding background, causing highly discrete values for that group.

ing the requisite videos (0.13). Additionally, when looking at students who completed prerequisite materials, completing finger exercises (0.11) was more highly correlated with success than watching the prerequisite videos (0.02). We consider the correlation between doing and assessment scores a sufficient basis for further analysis.

3.2.2 The Doer Effect by Prior Experience

We now wish to know how the doer effect manifests for students of different experience levels. In order to conduct this regression analysis, we use a similar mixed effects model as we did for all students, this time replacing each independent variable with six variables, one for each of the survey responses. Note that each student will have non-zero values only for the six coefficients corresponding to his/her survey response. The results of this regression are also shown in Table 3.2.2. In this case, we see some actions remain similarly correlated over different prior experience groups while others vary from group to group. For students with the least experience, who answered 'No Experience', videos and finger exercises have similar coefficients. For students with a moderate amount of experience, i.e. those who know another language and those who know Python, it seems that finger exercises were a relatively better indicator than videos of success. Finally, for those who identified as veterans, we see a similar pattern as those in students with no experience where videos and finger exercises are similarly correlated.

It may be possible that doing is strong indicator of success in only in students with a moderate amount of experience, whereas watching videos is comparatively more helpful at the extremes of the prior knowledge spectrum. A possible explanation for this behavior is that veterans draw on their own knowledge to guide their behavior, while those with no experience rely on the instructor's curriculum to guide their behavior. Both of these trajectories are similar, causing both veterans and novices to behave similarly to learn the material.

features	All		Know Python		No Experience		No Response		Other Language		Veteran	
	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value
prereq vid	0.02	0.04	0.09	0.00	-0.04	0.03	0.04	0.05	-	-	-	-
postreq vid	-0.05	0.00	-0.05	0.02	-0.04	0.05	-	-	-0.06	0.00	-	-
withinreq vid	0.13	0.00	0.05	0.00	0.18	0.00	0.11	0.00	0.14	0.00	0.18	0.00
prereq fex	0.11	0.00	-	-	0.19	0.00	0.13	0.00	0.09	0.00	0.18	0.00
postreq fex	0.04	0.00	-	-	-	-	-	-	0.04	0.00	-	-
withinreq fex	0.37	0.00	0.45	0.00	0.28	0.00	0.39	0.00	0.37	0.00	0.27	0.00

Table 3.2: Regression coefficients and p-values where problem set performance is a function of watching videos and completing problems over different time scales. We have two models here: one for all students ('All' on the far left), and another where students are differentiated by their prior experience level. We only show coefficients significant at the p=0.05 level.

features	All		Took 6.00.1x		Know Python		No Response		No Experience	
	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value
prereq vid	-	-	-	-	-	-	-	-	-	-
postreq vid	-	-	-0.06	0.05	-	-	-	-	-	-
withinreq vid	0.03	0.05	-	-	-	-	-	-	-	-
prereq fex	-	-	0.09	0.00	-	-	-0.07	0.01	-	-
postreq fex	-	-	-	-	-	-	-	-	-	-
withinreq fex	0.48	0.00	0.46	0.00	0.63	0.00	0.49	0.00	-	-

Table 3.3: Regression coefficients and p-values obtained using the same method for 6.00.2x. Our findings from the 6.00.2x data are much noisier, yet we still see the within-unit finger exercises being the strongest indicator of success.

3.2.3 The Doer Effect in Advanced Courses

With the doer effect very clearly related to student learning, we now wish to see how this varies over the advanced nature of a course. Specifically, we will consider 6.00.2x, as described in Chapter 1; this course is expected to be a natural progression of 6.00.1x and thus covers more advanced content. For this course, we use a similar mixed effects model as we did for 6.00.1x, with the results shown in Table 3.3.

We see that aside from the the within-unit finger exercises being most highly correlated with success among most groups, many of the other parameters are surprisingly noisy especially when compared with 6.00.1x. results. We believe there may be three reasons for this result. First, the number of enrolled students and problem sets are both higher in the introductory course; with its six problem sets and broader audi-

features	All		No Experience		Other Language		Know Python		Veteran		No Response	
	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value
vid	0.10	0.00	0.14	0.00	0.10	0.00	0.12	0.00	-	-	0.08	0.05
fex	0.35	0.00	0.35	0.00	0.42	0.00	0.32	0.00	0.37	0.00	0.25	0.00

Table 3.4: In the long term, we also see that finger exercises are more correlated with problem set success with the effect being three times that of videos.

ence, 6.00.1x contains 20,039 student-unit combinations while 6.00.2x only contains 3,312 due to fewer units (four) and fewer students. This difference could affect the stability of our regression. Second, we posit that the more advanced course material in 6.00.2x could lead to higher differentiation in the abilities of the survey respondents i.e. those who took 6.00.1x are at a different experience level than those who answered that they know Python. This may explain the stronger coefficients we see overall while muddling the results for all students. Third, we note that the magnitude of available material is quite different between the two as well. Whereas 6.00.1x has 555 finger exercises and 81 videos, 6.00.2x has 177 finger exercises and 43 videos. Though this difference may not seem large, it decreases the amount of differentiation among students when considering with-unit and outside-unit activity.

3.2.4 The Doer Effect in the Long Term

The models we have used thus far use problem sets as a proxy for learning; problem sets are completed at a student's own pace while the student also has access to finger exercises and videos. We consider this to be 'short-term' learning because students are assessed on material immediately after learning. We now wish to consider 'long-term' learning as well, where we consider final exam grades to be a proxy of this performance. In this case, we use a slightly different regression where we no longer need random effects because each student takes the final exam once, and there is only one type of exam, giving the following regression formula: `final_grade.Z` $\sim$ `num_fex.Z + num_vid.Z`. The results of this experiment are shown in Table 3.4.

Again we see that the doer effect is prominent, this time when considering how students perform on an assessment weeks after their the content is learned. In fact,

we see that doing is three times a stronger predictor of success than watching videos; we also see that this trend exists across all groups except veterans where watching videos is not a significant predictor.

3.3 Discussion

In order to evaluate the doer effect in a computational setting, we sought to mimic existing methods in order to provide as true a comparison as possible [13]. In general, we find that our results are in line with existing literature on non-computational subjects i.e. that the relative amount of doing is a stronger predictor of assessment success than watching videos. In fact, we saw this trend exist to some degree in the general computational learning case, for different experience levels, for more advanced courses, and for longer term learning. We do, however, note that in some cases, such as the 6.00.2x results shown in Table 3.3 and in the prior experience study in Table 3.2.2, the results are less conclusive; in the contexts of advanced study and prior experience, we believe more research must be done to have a conclusive result. Nonetheless, studying student behavior in the MOOC setting is unique because we have greater insight into how students behave than we would in a traditional classroom setting i.e. how exactly they interacted with videos and problems on a click-by-click basis. If one assumes that these actions are a better method of capturing 'doing', we can use this additional granularity to give greater insights into how the doer effect exists specifically in a computational MOOC setting.

Chapter 4

Integrating Enriched Features

Identifying 'doing' in a MOOC setting is challenging due to the varying definitions of doing practice problems. Because MOOCs collect more granular data regarding student behavior, we can be more specific about how we wish to quantify doing. In this chapter, we will model student behavior using specific actions that users can take on a MOOC platform to better understand doing and how it is correlated with learning. Instead of considering each problem and video as a task to be completed, we instead consider the problem solutions as either being checked or revealed and videos as a continuous value with a number of minutes being watched. In the finger exercises space, we believe that considering both of these actions will better capture the doer effect. In the video space, we believe that total minutes watched, including rewatching will similarly be a better measure of traditional learning modes. We seek to clarify the relationship between problem set and exam performance and the doer effect using these enriched features.

4.1 Method

In order to understand how the doer effect manifests in these courses, we've taken a similar approach to Koedinger et al [13, 12] with some modification, though we still use a linear regression model where we predict either problem set scores or final scores as a function of student activity.

4.1.1 Feature Engineering

We consider three features in our model of student behavior: minutes of video watched (video), number of times a problem was checked (check), and number of times a student can reveal a solution (reveal); these are the same features we present in Table 1.3. Because we want to relate assessment grades to activity within the same unit, we bin student activity into 'before-unit', 'within-unit', and 'after-unit'. For example, if a student watches a video corresponding to the Structured Types unit and another video on Object-Oriented Programming, and these videos were watched in week 9, in the week preceding the Object-Oriented Programming problem set, the first video would be considered "after-unit" and the second would be 'within-unit'. Note that this is a similar differentiation over time that the literature utilizes. After computing these values for all user-unit combinations, we then normalize the results on a per-unit basis to control for differences in the number of available activities for different units. Thus, for each of the features, 'video', 'show answer', and 'problem check', we have three time buckets: before, during, and after. We also normalize problem set grades per unit. Figure 4-1 visualizes this technique.

This approach differs from the previous method in two major ways.

First, we are considering actions done only in the week preceding the due date of a problem set as opposed to all actions since the beginning of the course. As shown in Figure 2-2, video watching among student spiked on problem set due dates, indicating that the week preceding the due date is most relevant for learning material. We additionally considered only including actions taken from the time a problem set was released to its due date, however we found that due to the overlapping time-frames over which assignments could be due, this was less informative.

Second, instead of counting the number of problems *attempted*, we consider the number of actions ('problem check' or 'show answer') taken. Previously, the maximum value for a 'within-req-fex' was the number of finger exercises available in the unit. Each finger exercises was either attempted in some regard or not at all. In this new regime, the number of problem checks is theoretically uncapped and we isolate the

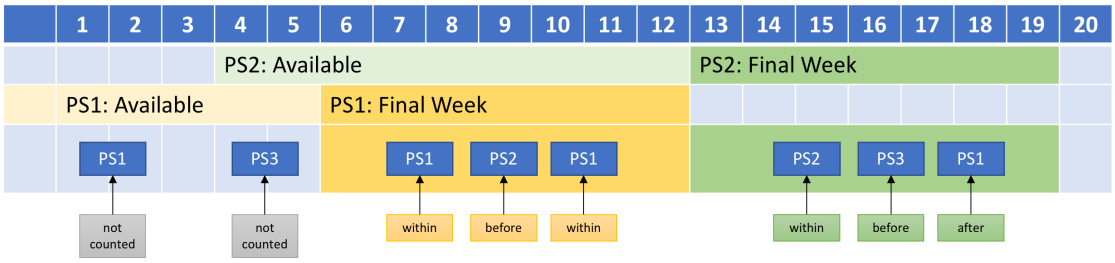

Figure 4-1: When looking at a calendar of the course's due dates and when a student takes certain actions, we can compute the corresponding features. The first row of the figure enumerates the days, while the second and third rows show the timespans for two problem sets with the first due on the 12th day and the second problem set due on the 19th day. Each of the eight boxes in fourth row then indicates a certain action taken by the student, with the corresponding unit noted in the box. For example, the first box on the left indicates a finger exercise or video for unit 1. Because the first two actions do not fall in the seven days prior to a due date, they are not counted. Of the three actions completed during the critical time for PS1, two are relevant to PS1, giving the 'within' designation, while the other is for PS2, giving the 'before' designation because the actions occurs before the unit has been covered. A similar classification occurs for the actions taken in days 13-19 before the PS2 due date.

effects of revealing a solution versus checking the problem. This is useful because in computational subjects, checking a problem can be helpful if one dutifully interprets the error message. Furthermore, our exploratory analysis in Figure 2-3 supports the idea that there exists variation in how students reveal solutions vs check solutions.

4.1.2 Regression

Whereas in the previous section, we used a mixed effects regression with random effects for the user and the unit of the problem set, we use only fixed effects here because we found that the mixed effects models had prohibitively high condition numbers, implying numerical instability in our resulting coefficients. Additionally, when considering the groups of students based on survey response, we opt to run a separate regression for each group instead of creating an effect for each variable in each group. The resulting formula would be used in R as follows:

```
lm(unit_grade.Z ~ video_within_unit + video_before_unit
+ video_after_unit + show_answer_within_unit + show_answer_before_unit +
```

`show_answer_after_unit + problem_before + problem_within + problem_after, data=edx_data)`.

Note that for all experiments conducted in this chapter, we used the Python statsmodels framework.[1]

4.2 Experiments and Results

We can evaluate our results by observing the significant regression coefficients produced by our analysis. Because our covariates are the z-scores of each type of student action, we interpret a positive coefficient as the grade being positively correlated with how much of a certain action a student takes relative to his/her peers. The magnitude of the coefficient then indicates how many standard deviations away from the mean assessment score we expect students to perform for each additional standard deviation from the mean of their actions.

4.2.1 The Doer Effect in Computational MOOCs

We begin by considering the the doer effect for all students in the first computational course, 6.00.1x, shown in Table 4.1. The leftmost columns indicates the coefficient values and significance for each action. Initially, we see that all the 'within' coefficients are of similar magnitudes, indicating that watching videos is as correlated with success as both checking and revealing solutions. We also see that watching videos after a unit has finished and checking problems for a unit after the unit has finished are both negatively correlated with success (check after (-0.05) and video after (-0.06)). Just as certain actions taking place after the due date is negatively correlated with success, watching videos ahead of time ('video before') is the most highly correlated action (0.22), indicating that students who start learning material more than a week in advance tend to perform well.

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

	All		Know Python		No Experience		No Response		Other Language		Veteran	
features	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value
video within	0.08	0.00	-	-	0.06	0.00	-	-	0.05	0.00	0.16	0.03
video before	0.22	0.00	0.19	0.00	0.26	0.00	0.21	0.00	0.21	0.00	0.21	0.00
video after	-0.06	0.00	-0.06	0.03	-0.10	0.00	-	-	-0.04	0.02	-0.50	0.00
reveal within	0.09	0.00	0.07	0.00	0.11	0.00	0.07	0.02	0.07	0.00	-	-
reveal before	0.02	0.04	-	-	-	-	-	-	-	-	-	-
reveal after	-	-	-	-	-0.05	0.02	-	-	-	-	-0.25	0.00
check before	0.02	0.02	-	-	-	-	0.09	0.02	0.04	0.03	-	-
check within	0.06	0.00	0.13	0.00	0.13	0.00	0.19	0.00	0.14	0.00	-	-
check after	-0.05	0.00	-	-	-	-	-	-	-0.06	0.00	0.45	0.00

Table 4.1: This regression models problem set grades as a function of student activity for 6.00.1x problem sets. Each row is a different affordance/time-frame for which a user can complete activities for a unit. The columns indicate different subsets of students based on their prior experience indicated in Table 1.3, with the first column describing all students. An activity is considered relevant if it is completed in the seven days preceding the problem set due date. Only coefficients significant at the p=0.05 level are included.

	Know Python	No Experience	No Response	Other Language	Veteran
video within	-	0.75	-	0.63	2.0
reveal within	0.77	1.22	0.77	0.77	-
check within	2.17	2.17	3.17	2.33	-

Table 4.2: We measure the strength of certain actions for certain groups of prior experience groups by expressing $R_s^f = \frac{\beta_s^f}{\beta_{all}^f}$ for each prior experience group and for each of the within-unit actions. Higher ratios indicate that this action is more highly corrrelated with success compared to the general student cohort. We only list coefficients where they are statistically significant. Note that watching videos is far stronger for veterans compared to the general population and checking solutions is not even relevant for veterans but strong for all other groups.

features	All		Know Python		No Experience		No Response		Other Language		Veteran	
	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value
video within	0.08	0.00	-	-	0.01	0.00	0.01	0.04	-	-	-	-
video before	0.22	0.00	0.02	0.00	0.04	0.00	0.04	0.00	0.03	0.00	-	-
video after	-0.06	0.00	-	-	-0.01	0.00	-0.02	0.00	-0.01	0.01	-0.04	0.00
problem check within	0.09	0.00	0.03	0.00	0.02	0.00	0.04	0.00	0.03	0.00	-	-
problem check before	0.02	0.04	-	-	-0.01	0.00	-	-	-0.01	0.04	-	-
problem check after	0.00	0.00	-	-	-	-	-	-	-0.01	0.03	-	-
show answer within	0.02	0.02	0.01	0.00	0.01	0.00	-	-	0.01	0.00	-	-
show answer before	0.06	0.00	-	-	-	-	-	-	-	-	-	-
show answer after	-0.05	0.00	-	-	-	-	-	-	-	-	-	-

Table 4.3: To offer an alternate method of comparing the strength of doing among the student groups, we used a linear model similar to that in Chapter 3; we have 45 coefficients, nine for the features across the five survey groups. We also include mixed effects for unit and user. We see that it is more difficult determine differences among the covariate coefficients due to similarities in their magnitudes and the overlap in the 95% confidence intervals (not shown).

4.2.2 The Doer Effect by Prior Experience

We then consider how context of prior experience affects the manifestation of the doer effect. In particular, we seek to understand how different actions are correlated with success for students of different experience groups.

Using Coefficient Ratios to Compare Prior Experience

To understand this effect, we examine the ratios of coefficients for the experience groups in Table 4.1. A ratio of the coefficient for a covariate for all students to the same coefficient for a subgroup informs us of the effect of that covariate for a specific group. Thus, we can characterize the effectiveness of a particular feature R_s^f where f is the feature and s is the survey subgroup as follows:

$$R_s^f = \frac{\beta_s^f}{\beta_{all}^f}$$

The coefficient value for each action and each prior experience group is listed in Table 4.2. Using this metric, we see that $R_{veteran}^{video,within} = 2$ whereas $R_{none}^{video,within} = 0.75$, indicating that watching videos is more highly correlated with success for veterans than for students without experience. Likewise, we see that for all survey groups

features	All		Took 6.00.1x		Know Python		No Response	
	coef	p-value	coef	p-value	coef	p-value	coef	p-value
video within	-0.10	0.00	-0.14	0.00	-0.25	0.01	-	-
video before	0.16	0.00	0.14	0.00	-	-	0.16	0.00
video after	-0.07	0.00	-0.09	0.00	-	-	-	-
reveal within	0.17	0.00	0.17	0.00	0.25	0.01	0.13	0.00
reveal before	-	-	-	-	-	-	-	-
reveal after	-	-	-	-	-	-	-	-
check before	0.04	0.01	0.08	0.03	-	-	-	-
check within	0.06	0.00	0.14	0.00	-	-	0.15	0.00
check after	-	-	-	-	-	-	-	-

Table 4.4: These regressions utilize the same model as those in Table 4.1, though for 6.00.2x. The corresponding groups are for 6.00.2x, for which the survey responses were slightly different. Note that while a 'no experience' option existed in the survey, only 2 of these students completed the course, so a meaningful regression was not possible. Again, only coefficients significant at the $p = 0.05$ level are included.

except veterans, $R_s^{check,within} \sim 2$, but for veterans, checking problems within the unit is not a significant covariate [2]. This relationship indicates that for students with prior experience, doing is less correlated with success, whereas videos are more correlated with success.

An Alternate Prior Experience Comparison

We additionally offer an alternate method of comparing coefficients among the different groups. In this case, we use a similar method as that displayed in Table 3.2.2, though using the feature engineering method used above. The results of this analysis are shown in Table 4.3. We see that it is more difficult to make distinctions in the power of the covariates for the different groups. Using this method, checking problems within unit is highest for students with some experience i.e. those who know Python or another language, compared to those with no experience. We do, however, note that this is not a strong difference and more analysis may be required.

[2] We note that relatively few veterans completed the course, which may bias our estimates.

4.2.3 The Doer Effect in Advanced Courses

We now consider whether the material of a subject influences the presence of a doer effect i.e. is the doer effect present in a course with more advanced material. To address this question, we consider a second course on edX, 6.00.2x, Introduction to Computational Thinking and Data Science. This topic typically has fewer students enrolled with four units instead of six. On edX, the expected trajectory is for students with no experience to complete 6.00.1x and then 6.00.2x. We replicated our method for the more advanced course, shown in Table 4.4, and in this case, we will compute a statistic Q^c_{adv} – the ratio of the magnitudes of regression coefficients where c is the course:

$$Q^c_{adv} = \frac{\beta^c_{reveal}}{\beta^c_{check}}$$

We can now compute $Q^{6.00.1x}_{adv} \approx 1.5$ and $Q^{6.00.2x}_{adv} \approx 4$ in order to compare the relative strength of correlations for different courses. We thus see that the 'show answer' feature is much stronger in the more advanced course; students who elect to reveal solutions more often than the average student fare well. We also note that watching videos is actually negatively correlated with success in the more difficult course, indicating that the more time a student spends watching videos above the mean video watching time, we can expect problem set score to decrease. We have also included the coefficients for different survey response groups. We see that both for students who know Python and those that took 6.00.1x, most of the effects seen for all students hold true. This could indicate that for more conceptually advanced material, prior experience factors less into performance.

4.2.4 The Doer Effect in the Long Term

The final question when considering the presence of the doer effect was time: does the doer-effect exist when students are assessed on material in the short-term and the long-term. In the courses we analyze, students' grades are determined by performance on problem sets for each unit as well as a final exam at the end of the term. Because

	All		Know Python		No Experience		No Response		Other Language		Veteran	
features	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value
check	0.03	0.00	-	-	-	-	-	-	0.05	0.00	-	-
reveal	-0.08	0.00	-	-	-0.09	0.01	-	-	-0.05	0.04	-0.26	0.03
video	0.21	0.00	0.24	0.00	0.23	0.00	0.23	0.00	0.20	0.00	0.29	0.02

Table 4.5: This regression models problem set grades as a function of student activity for the 6.001x final exam. On the left are the different (learning mode, time scale) combinations for which students can complete activities for a particular unit. The columns indicate different subsets of students based on their prior experience, with the first column describing all students. Because we are modeling the final exam, there is no distinction for "within-unit" activities, and the student's activity over the entirety of the course is used to construct features. Only coefficients significant at the p=0.05 level are included.

the final exam concerns material from the entire course, we use it a proxy for long-term learning. In order to compare final grade coefficients with problem set coefficients, we again define a ratio to apply to each regression, S_m where m is the measure of time, either short (measured by problem sets) or long (measured by final exams):

$$S_m = \frac{\beta^m_{video}}{\beta^m_{check}}$$

In this, case, $S_{short} \approx 1.33$ whereas $S_{long} \approx 7.0$, indicating that when completing the final, the relative number of videos watched over the term is a better indicator of student performance than completing problems. The discrepancy in the ratios tells us that when learning material in the long term, the doer-effect tends to give way to watching videos. We also note, with input from the course instructor, that final exam problems are more general and require less mastery of the content of the course than the individual problem sets.

4.2.5 The Doer Effect for Specific Topics

Learning computational thinking encompasses many skills, Many of which are quite different from traditional modes of cognition [21]. Additionally, different topics within computer science are also quite different such as search/sort algorithms and object-oriented programming. In order to understand how the doer effect manifests across

features	Bisection Search coef	Recursion coef	Object-Oriented Programming coef	Complexity coef
video	0.01	0.14	0.42	0.44
reveal	0.07	-0.08	0.17	0.24
check	0.31	0.09	0.22	0.26

Table 4.6: To better study specific subjects, we formulate our regression problem to predict student grades on problems in a problem set that assess a certain topic; the features, are constructed similarly as for the previous results, but here, we only select videos that correspond to the topic as well as specific finger exercises that also correspond to the topic. Thus while the content for a particular unit may include many topics, we seek to isolate specific topics. All coefficients listed are significant at the p=0.01 level

different topics, we studied four topics recommended by the 6.00.1x instructor: recursion, object-oriented-programming, complexity theory, and bisection search. We used a similar regression model, though with slightly different predictors. Instead of considering all videos watched/exercises attempted, we only consider material that covers the topic of interest.

From the results, shown in Table 4.6, we find that the correlation between different learning modes and student performance is highly variable over different topics. We start by examining the coefficients for videos which are almost negligible in the case of bisection search (0.01), but highly correlated with succeeding in object-oriented programming (0.42) and complexity (0.44). Likewise, checking solutions was similarly related to performance for complexity, object-oriented programming, and bisection search ($\sim$0.26), but much less so for recursion (0.09). Finally, we see that revealing solutions is correlated with learning quite differently as well, even being negatively correlated with assessments on recursion (-0.08). When considering these results, we do take into consideration that certain topics, such as recursion did not show up on the final exam, perhaps deincentivizing learning the concepts. Additionally, the number of exercises available to students for each topic varies; for example, students were not given ample exercises to learn complexity, perhaps changing their behavior when learning the underlying concepts.

4.3 Discussion

The doer effect, while definitely present, is not without caveats. We saw that doing is at least as effective as watching videos for the general student population, but this relationship broke down when we considered students of different experience levels. Instead, we saw that while doing was still correlated with success for most groups of students, for the 'veterans,' watching videos was uniquely correlated with success. We then considered the more difficult 6.00.2x course, where we saw that doing is still prevalent albeit in a different form. Students who elected to reveal solutions were doing comparatively better than those who took the same action in the more basic course. When considering the long term and the short term, we saw a stark difference in the relationship between performance and doing. While doing and watching are similar in the short term, watching videos proved to be a much better indicator of long-term knowledge. Finally, we consider the doer effect for a smattering of different canonical topics, finding that the doer effect varies as one may expect: Concepts like object oriented programming appear to be learnable via trial and error on finger exercises whereas more nuanced concepts like complexity seem to be better approached with videos.

These findings both support and contradict our results from Chapter 3 when we utilized methods from prior works [13]. Some results were found regardless of the method i.e. that within-unit doing is correlated with higher assessment grades for all students in the standard course (6.00.1x) as well as in the more advanced course, 6.00.2x. Where the methods produce different results is when we consider prior experience. Using the work presented in the literature, we saw that watching videos is correlated with success for novices and veterans, while those in the middle of the knowledge spectrum used finger exercises. However, using the enriched definitions of doing, we saw that finger exercises were correlated with success in all students *except* veterans. It is possible that the results presented in Chapter 3 do not differentiate the veterans because the veterans are accessing all the materials like other students. Where they may differ in their actions is the actual number of minutes watched

or number of times solutions are checked. In this case, the enriched features may give veterans more space to differentiate their actions compared to other students, validating the results we saw in this chapter.

Chapter 5

Predicting Prior Experience in a MOOC

A MOOC cohort will often have students with varying experience levels. Our work in previous chapters shows that students who self-identify as "veterans" may have different behaviors than other students who are unfamiliar with the material. Being able to isolate these students in a data set containing hundreds or thousands of students allows us to make stronger conclusions from our data; additionally, identifying veterans in a live MOOC setting can also be informative to instructors in a course with thousands of students. When students enroll in a MOOC, we can collect basic demographic information, such as age, level of education, and country. We can also observe a student's behavior on the MOOC platform such as their video-watching tendencies. Using this information, we wish to develop a classification model for veterans in a MOOC setting.

5.1 Method

Classification is well-studied topic in machine learning, with different methods offering different advantages. Logistic regression is one such method that often performs well, but is agnostic to prior information about our data, and prevents us from exploiting structure in our data such as related measurements.

5.1.1 Hierarchical Models

A canonical manifestation of these shortcomings is Gelman et al. 2017's Radon data set, where the level of radon in a home was measured in many homes in many different counties across the US; the data contains measurements for many homes in some counties and sometimes very few measurements per county. There are thus two ways to model the radon levels in a home. First, we can assume that all counties are identical and estimate a single regression for all the measurements in the dataset (pooled model). Alternatively, we could assume that counties share no similarities and estimate a regression for each county (unpooled model). However, neither of these models are realistic. Bayesian Hierarchical Models allow us to share information about different subsets of data, by assuming that all county regression coefficients are drawn from the same distribution [9].

Due to the nature of our data, where we have both demographic and behavioral information available for individual students. We will explore how a hierarchical model can improve our classification efforts while providing a greater degree of flexibility in fitting our model as well as being more interpretable across our data.

Due to improved, off-the-shelf samplers becoming more robust, Bayesian modeling is become more viable because more complex posterior distributions can be better approximated. NUTS, which adds adaptive tuning to Hamiltonian Monte Carlo is an often used sampling method when integration is computationally infeasible [11].

Bayesian methods are especially popular in creating hierarchical generalized linear models (GLMs). GLMs have three components: a probability distribution of the predictor i.e. $Y \sim N(\mu, \sigma^2)$, a linear predictor i.e. $\mu = X\beta$, and a link function: that relates $E[Y]$ to the linear predictor, i.e. $\log E[Y] = \mu$. This format generalizes well to many models such as logistic regression by using a logarithmic link function [8, 4].

Hierarchical models also enjoy a unique place in machine learning and data analysis by allowing us to exploit structure in our data such as certain data points being more similar or a natural nesting of parameters in our data. Adding a level of hierarchy (as described in the radon example) to a GLM can help model some of the

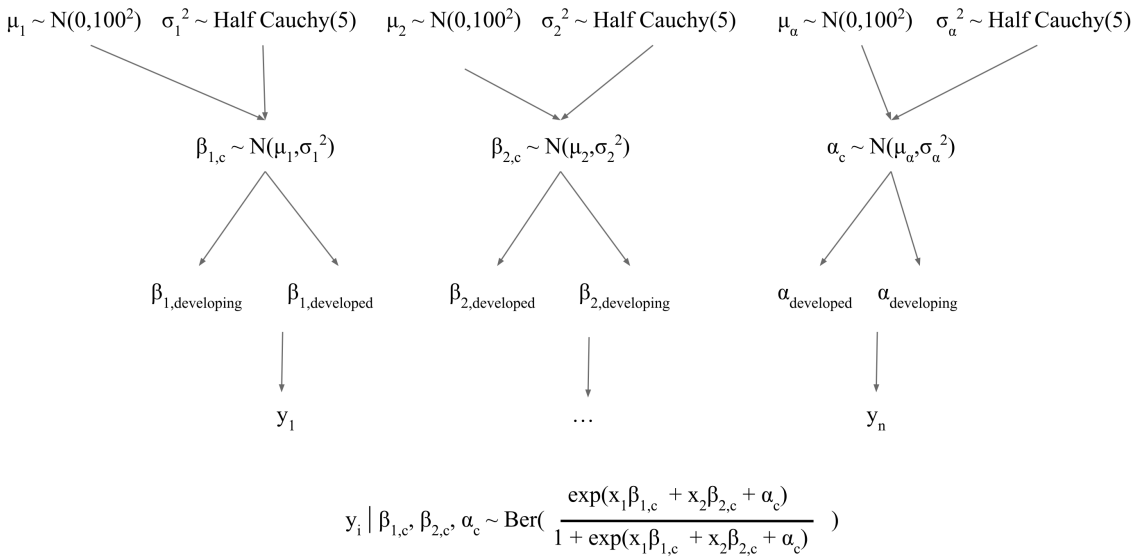

Figure 5-1: This figure defines one hierarchical model that we evaluated where we consider two variables in our logistic regression for which we have two coefficients β_1 and β_2 and an intercept α. In this model, we assume that these coefficients are independently drawn from normal distributions $N(\mu_1, \sigma_1^2)$, $N(\mu_2, \sigma_2^2)$, and $N(\mu_\alpha, \sigma_\alpha^2)$. Regression coefficients/intercepts are drawn once for each class. Here, we show partial pooling over the possible economic statuses ("developing" or "developed") of the countries of the user. The veteran status of each user is therefore a Bernoulli random variable with $p =$ the inverse of the logit of $X_1\beta_1 + X_2\beta_2 + \alpha$.

clustering effects that occur by introducing partial pooling.

The particular generative model that we propose is based on a combination of the demographic and behavioral information

5.1.2 Data

We will be working with data from the edX platform over two terms of 6.00.1x, Introduction to Computer Science and Programming Using Python. For each student, we collect both biographical data and behavioral data from the course itself, which are described in Table 5.1. Cleverly manipulating these features can allow us to discern which users are veterans and which are learning material for the first time. Although tens of thousands of students enroll in the course, only a few thousand complete the course. Students optionally complete a survey at the beginning of the course indicating their prior experience with the material. A small fraction of the students

Demographic Information	Behavioral Information
Gender	# pause video
Level of education	# seek video
UN Economic Group	# seq goto
UN Major Region	# unique videos viewed
Year of birth	# videos viewed
	# show answer
	# problem check
	# chapters
	# play video
	# days active
	# events

Table 5.1: We use a mix of demographic and behavioral features to classify students. The 'UN Economic Group' tells us whether the user's country is classified by the UN as 'developed' or 'developing'. Likewise, the 'UN Major Region' tells us the region such as 'Western Europe,' 'Southern Asia,' etc, which is slightly less granular than the user's actual country or even city. Additionally, we map education levels to the equivalent years of schooling i.e. five for elementary school, twelve for high school, sixteen for undergraduate. Behavioral information is collected from the entirety of the student's enrollment in the course; features generally count the number of times a student took an action on the edX platform such as pausing a video or revealing a solution.

complete the survey and a small fraction of these students identify themselves as "veterans". In order to balance the data set, we considered a random sample of students equal to the far fewer number of veterans and then used a 80% - 20% train-test split.

5.1.3 Models

We start with the baseline model: a non-Bayesian logistic regression model based on the features listed in Table 5.1. The non-Bayesian model does not consider any similarities across different hierarchies in the data, instead treating every variable to be of the same level.

We then consider a hierarchical model where we partially pool regression coefficients over students whose countries are of the same economic development status. The motivation for partial pooling over the economic status of the country is two-

fold. First, the coefficients for features such as age and level of education may be different in countries with different development statuses. Second, it is likely that among each of these subgroups the training data is unbalanced i.e. there are fewer students coming from developing countries, leading to more biased estimates.

This trade-off between needing to share information between groups while having minor differences motivates the need for a partial-pooling hierarchical model. Note that while we would pool over these demographic variables, we would consider behavioral variables independent of these hierarchies and would estimate one coefficient for the entire set of training data for behavioral features. That is, we would consider all students, independent of the partial pooling cluster to have similar coefficients dictating actions like watching videos or completing problems. We have illustrated the hierarchical portion of the corresponding graphical model in Figure 5-1.

We then consider a similar model, where instead of partial pooling over economic groups, we partially pool over geographic regions. The motivation here is that pooling over regions is either as informative or more informative than doing so over economic groups because individuals from the same region will be more similar than those from the same economic group.

5.2 Experiments and Results

Our baseline is a standard logistic regression model with 122 data points and an 80-20 train-test split – results shown in Table 5.2. Examining the odds ratio for the coefficients makes intuitive sense; when evaluating our models, we want to not only have high predictive accuracy but also understand what the coefficients say about the data. In this case, we will consider the predictive accuracy in a subsequent section.

Before examining our results on edX data, we present a hierarchical model run on simulated data for which the true distribution is known. In this case, we generated $x_{1,i}, x_{2,i} \sim N(0, I)$, and for half the data, we generated a Bernoulli random variable $y_i \sim Ber(logit^{-1}(6x_1 + 3x_2 + 6))$ and $y_i \sim Ber(logit^{-1}(-4x_1 + 3x_2 + 6))$ for the other half. We then created a hierarchical model with partial pooling for β_1. The trace of

Coefficient	Value	Odds ratio
LoE	0.009	1.01
YoB	-0.638	0.52
npause_video	-0.324	0.72
nseek_video	0.3231	1.38
nseq_goto	-0.0644	0.937
nvideos_total_watched	-0.0668	0.935
nvideos_unique_viewed	-0.0668	0.935
nshow_answer	-0.4003	0.67
nproblem_check	0.0045	1.01
nchapters	-0.1686	0.85
nplay_video	-0.1109	0.900
ndays_act	-0.2535	0.776
nevents	0.3725	1.45

Table 5.2: We can understand our regression coefficients by considering $\exp \beta$, which gives us the odds ratio if a variable increases by 1. Observing some of the values, we see that having more events i.e. clicks, substantially increases the odds that a user is a veteran as does seeking out specific points in the course videos. We also see that higher years of birth decrease odds of a veteran, which is also sensible (higher birth year = younger).

the posterior is shown in Figure 5-2. In this case, we see that the sampled posterior does in fact correctly approximate the coefficients used to generate the data, with the modes for β_1 occurring very close to the true values. While we did not expect the posteriors over α to be so different, we are more concerned with the coefficients than the intercept because they tell us more about the strength of different predictors in each of the pools. Finally, we see that the modes of β_2 are close to the true value, indicating that the hierarchical model accurately infers the parameters.

We now consider the edX data, first looking at both an unpooled (traces in Figure 5-4) and partially pooled (traces in Figure 5-3) model.

Comparing these two models, we see a few major differences. First, without the imposition of a prior on the coefficients of the unpooled data, we see smoother posteriors over possible values, whereas when using a partial-pooling model, the posteriors look less like the normal distribution we expect. Nonetheless, it is clear that the distributions over the coefficients and intercept are similar in both cases. One reason for this is the uninformative prior that we imposed in the partial pooling model that

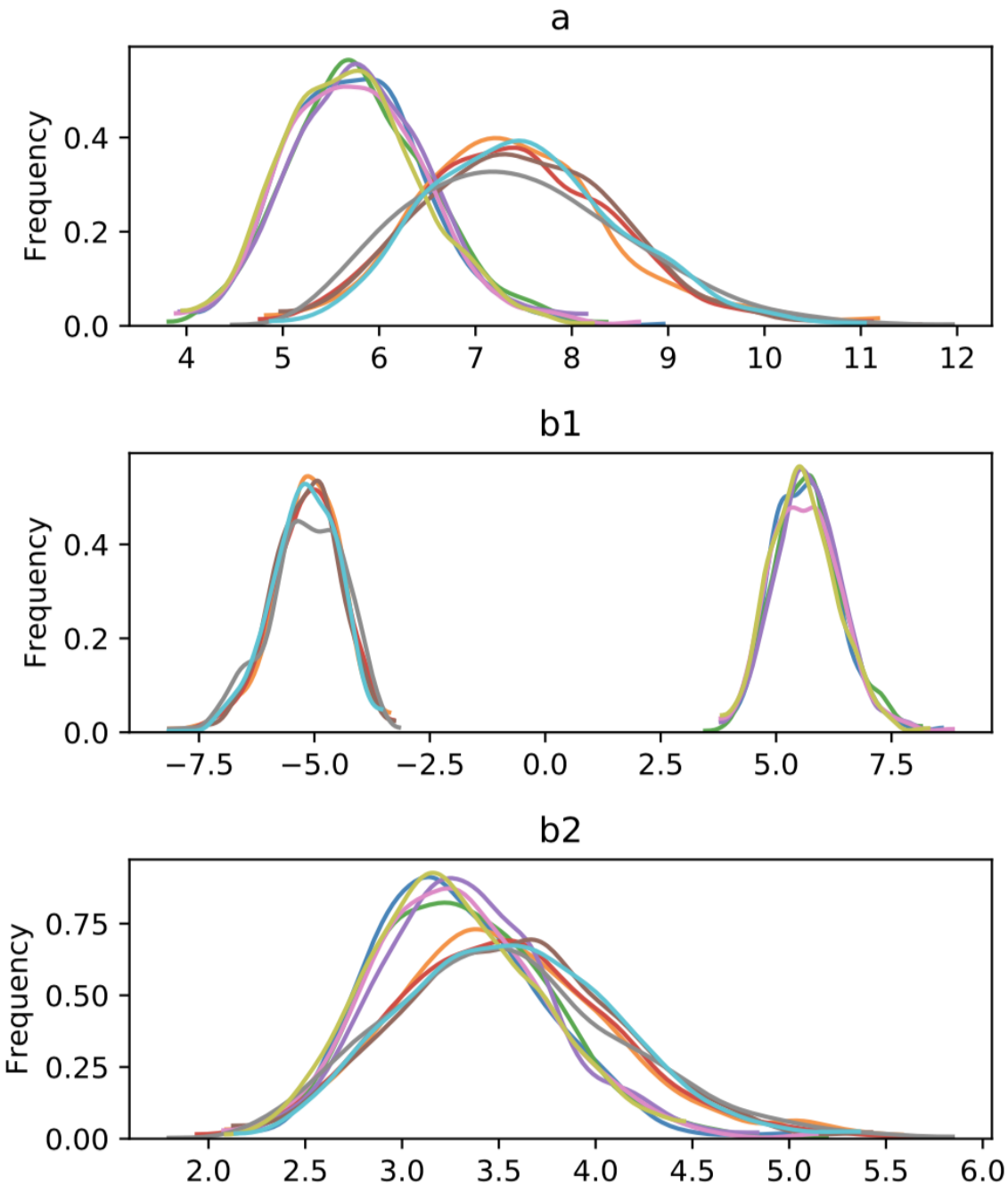

Figure 5-2: A unit test trace of simulated data where two different values of β_1 were used to generate data. The simulated data indicates that the model works as intended with the posterior distributions of β_1 falling on the two simulated values. We would create two classifiers, one where β_1 is equal to the mode of each of the traces.

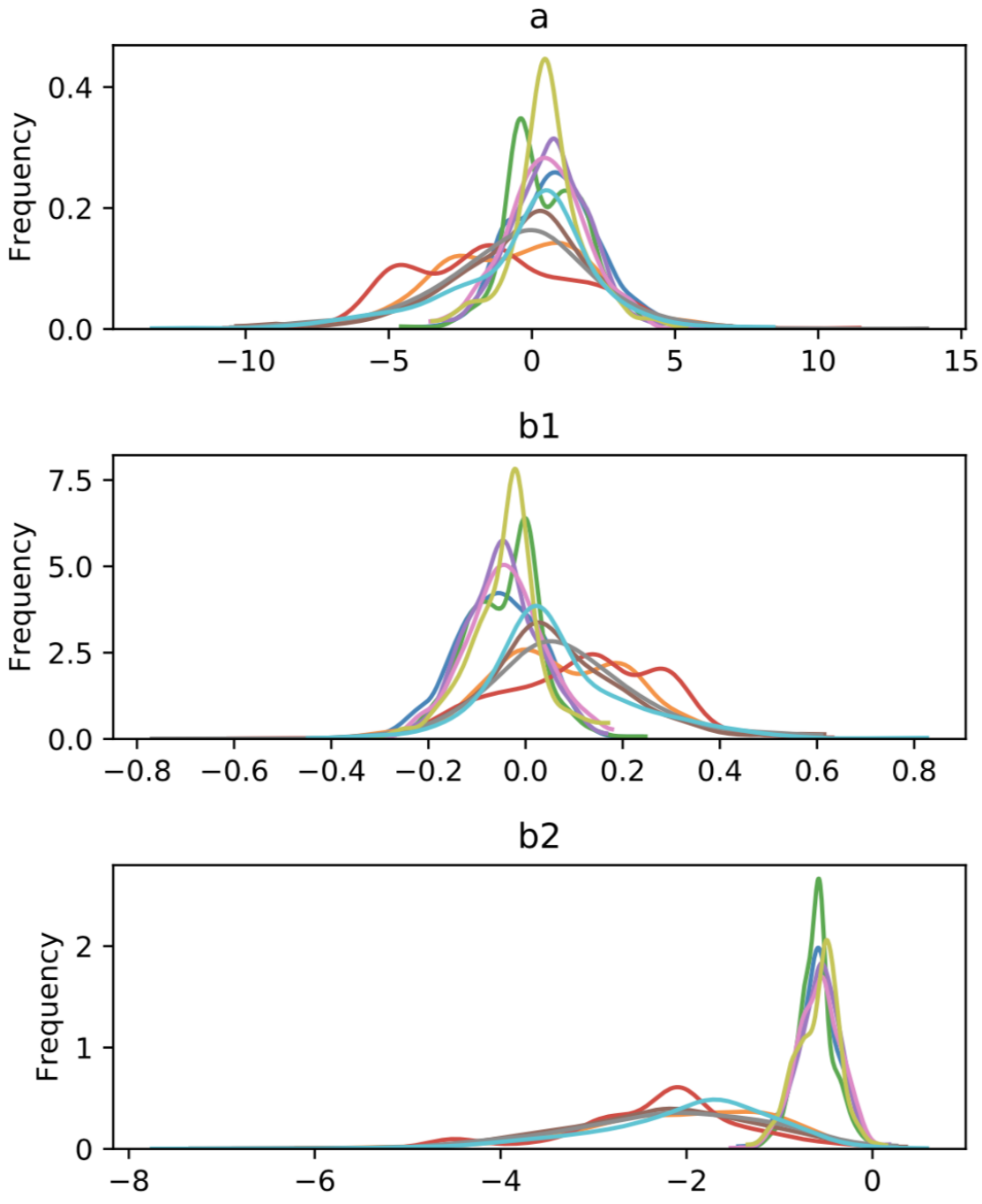

Figure 5-3: The trace plots represent posterior distributions for the logistic regression coefficients of a model that is partially pooled on economic group. The flatter distributions correspond to developing economic groups. β_1 corresponds to a student's level of formal education and β_2 corresponds to a student's year of birth. We see two posterior distributions because we've created a two dimensional Gaussian distribution over each coefficient to allow for an estimate over each class of data. The traces are noisier beacause they must fit the data in each class while still being constrained by the priors on the parameters.

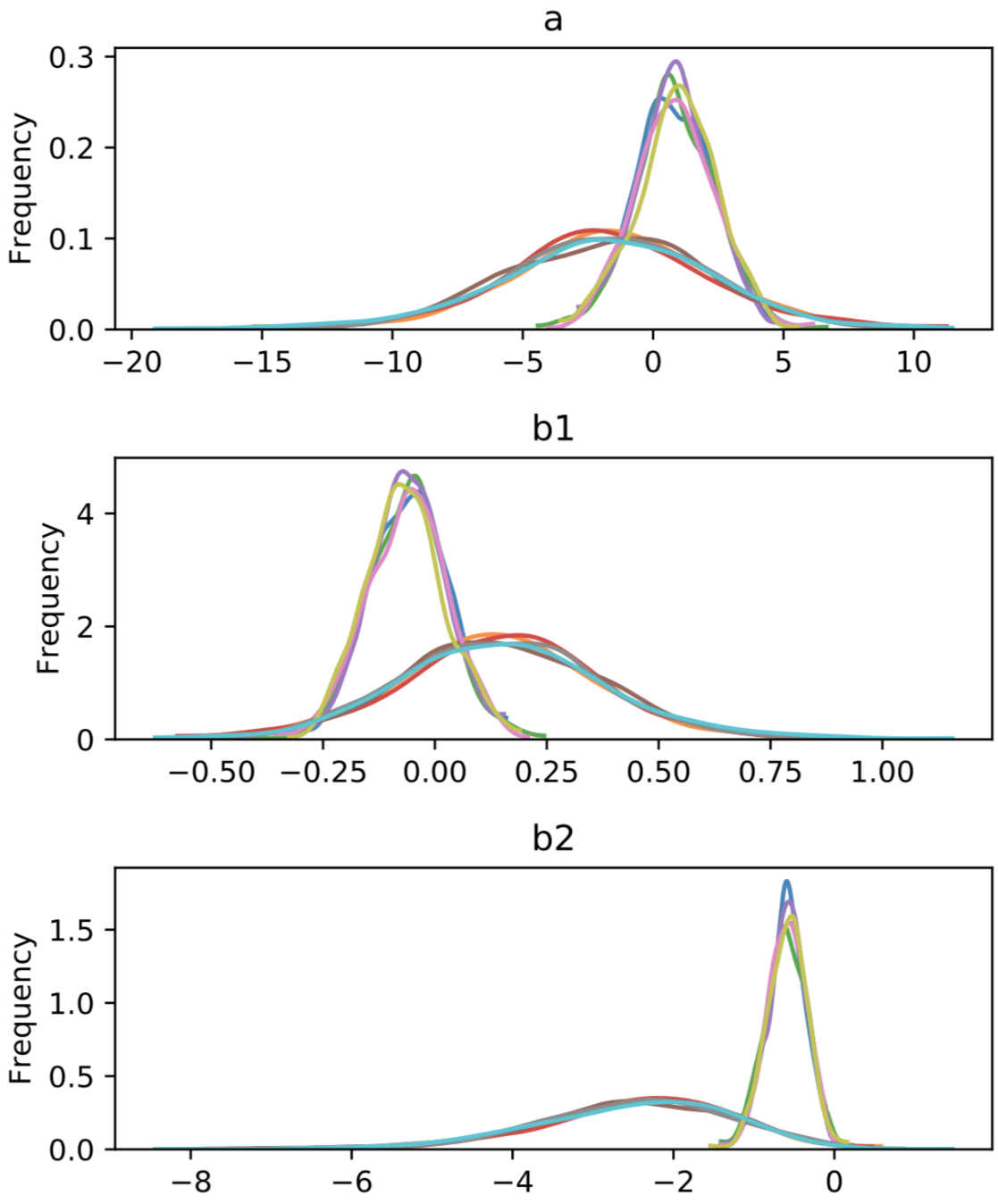

Figure 5-4: The trace plots represent posterior distributions for the logistic regression coefficients of a model with unpooled data i.e. each economic group treated independently. The flatter distributions correspond to developing economic groups. β_1 corresponds to a student's level of formal education and β_2 corresponds to a student's year of birth. While this is similar to the trace plots for the partially-pooled model, we see smoother estimates of the posterior because we are no longer constraining the coefficients for each class to be drawn from the same distribution.

allowed the posteriors on coefficients to take on any mean and length-scale[7, 17]; had we had a more informative prior, we could have expected more shrinkage in the partial-pooling model. We also see that these parameters are similar to those obtained in the logistic regression baseline. For example, the odds ratio in all three cases is greater than 1 for level of education but less than 1 for year of birth.

5.2.1 Other pooling methods

In addition to partially pooling users on the economic status of their country, we could partially pool on other factors, i.e. the country itself or the geographic region of the country. One can imagine that partial pooling on the country itself would be perhaps more effective than on the economic status because we have more granular information in our hierarchy. However, the most pertinent issue with this model is a lack of data. Creating a meaningful hierarchical model requires us to have a reasonable amount of data for each cluster in the hierarchy. While we can always use the hyper parameter priors to inform the posterior of a cluster with few data points, this would indicate that our partial-pooling method was too granular to begin with.

In addition to partial pooling on the economic group, we also compared partially pooled and unpooled analysis on the region of the country of the learner. One benefit was that although the unpooled model did not converge because in many countries there was not a single veteran enrolled in the course meaning the logistic regression model could not converge, the partially pooled model was able to create a posterior for each region, though of course regions with limited data were very similar.

5.2.2 Shrinkage

One beneficial aspect of partial pooling is additional information for clusters with fewer data points where our estimates are more biased. In this case, shrinkage will bring parameters over the hierarchical clusters closer together. For example, referring to Gelman's Radon data set, if some counties have over 30 measurements while others have only 1 or 2 measurements, partial pooling allows us to have a less biased estimate

Economic Group	# Students
Developed	92
Developing	30

Table 5.3: While our test set is balanced, the number of veterans in developing countries is far fewer than that in developed countries.

for counties with fewer measurements. Upon closer inspection of our data set, we see that unpooling our data based on economic region was problematic due to this lack of data, indicated by Table 5.3. The imbalance of data among these groups is important because this causes the posterior over the developing regions coefficient in the unpooled estimates to have a higher variance. However, when using the partially-pooled estimates, we would anticipate that despite having fewer observations in the developed regions, we can still obtain reasonable estimates of these values specifically because of the parameter sharing [19].

In order to visualize parameter shrinkage, we've shown how the partially-pooled model shrinks the parameter space by plotting both models in the β_1-β_2 space. We have included a plot comparing pooling to partial-pooling and an unpooled method to a partial-pooling method in Figure 5-6 and Figure 5-5.

5.2.3 Accuracy

One of the main metrics of any classification problem is the accuracy of the classifier, shown in Table 5.4. Some caveats here are that having more data would make these results more robust, and while we have constructed a balanced data set to both train and test our model to avoid the problem of class imbalance, the real distribution over veterans is highly imbalanced with only 2-3% of enrolled students identifying as veterans. The performance of the Partially Pooled Major Region Model tells us that by dividing our data by region and partially pooling the clusters, we can obtain higher accuracies than with a standard logistic regression model.

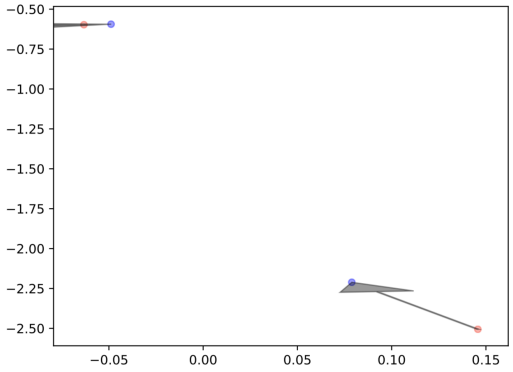

Figure 5-5: The x-axis is β_1 and the y-axis is β_2. The red scatter plot represents the model parameters over the two clusters in the economic groups when unpooled. The blue plots represent the economic groups when pooled. The arrows show the movement in the coefficient space, showing how the parameters exhibit some shrinkage when partial pooling is implemented.

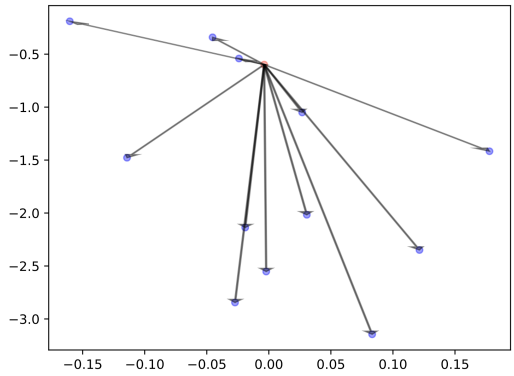

Figure 5-6: The x-axis is β_1 and the y-axis is β_2. The red scatter plot indicate the coefficients for each region with full pooling, and the blue scatter plot indicates the coefficients when partial-pooling is used. Here we see that partial pooling allows for the regions to take on very different parameters. Regions with less data cling to the mean

Model	Accuracy
Unpooled Economic Group	0.67
Partially Pooled Economic Group	0.67
Unpooled Major Region	0.62
Partially Pooled Major Region	0.76
Baseline	0.72

Table 5.4: We indicate predictive accuracies on a balanced test set of size $n = 14$. We see that the accuracy is similar across all models with a model Partially Pooled on Major Region giving the highest accuracy.

5.2.4 Convergence

The main statistic that we used to evaluate whether or not our NUTS sampler converged was the Gelman-Rubin method i.e. $\hat{R}$ is the ratio the average variance of the chains to the variance of all chains combined [3]. Literature as well as the PyMC library suggest that if this value is close to or greater than 1.01 then there may be an issue; we found that for all reported results, this was not an issue. While literature on monitoring the convergence of MCMC methods suggest using multiple metrics to evaluate whether a divergence occurred, reasonable accuracy on test data along with intuitive signs on coefficients seemed to indicate that the sampling was reasonable. For all results we used a burn in of 1000 samples and collected 4000 samples.

5.3 Discussion

Using hierarchical Bayesisan logistic regression allowed us to take a different approach to this problem. We found that by partially pooling our data on economic groups and major geographic regions, not only do we end up with posterior distributions that give us an idea of our confidence in the slope of our logistic curve, but we also get a more interpretable model of the nuances of our data, for example how certain parameters may differ over different pooling clusters as well.

We posit that this hierarchical model informed by some socioeconomic hierarchy of the student's location can be very powerful in modeling students in the MOOC setting where students come from very different backgrounds (both academically and geographically).

While this model proves useful in addressing the idea of identifying veterans, we believe that a more granular partial-pooling method, such as the actual country of the user combined with more informative priors to increase shrinkage may give a better model, along with of course, more data.

Chapter 6

Conclusions & Future Work

Through this work, we have considered the doer effect across many contexts. We employed data from two courses on the edX platform: 6.00.1x, Introduction to Computer Science and Programming Using Python, and 6.00.2x, Introduction to Computational Thinking and Data Science. We implemented a model similar to that presented in the literature [13], and we found similar results among most cohorts of students: The doer effect was consistently highly correlated with high performance on assessments. We then refined this model by allowing for greater granularity in our feature selection by redefining our measures of doing and video-watching. Considering the same cohort of students, we found similar results in many cases, though when considering the prior experience of students, our results differed. While the methods presented in the literature found 'doing' to be correlated with success for novices and veterans, our more granular method found watching videos to be uniquely correlated with doing for veterans. We posit that our more precise feature engineering may highlight the subtler behaviors of veterans that lead to this result. To further examine prior experience, we developed a classifier to predict the veterans in a MOOC cohort. Our logistic regression model uses partial-pooling to accommodate smaller, imbalanced data sets and achieve better accuracy than a standard logistic regression.

In future work, we will continue to improve our model of student behavior. Considering the prior experience prediction method, we will refine the model by using more data and accounting for class imbalance. We will modify our definition of 'doing'

to better differentiate student behavior. We can use our prior experience prediction methods to identify the prior experience level of students who we do not have prior experience information to obtain better estimates of the doing and video-watching coefficients. Finally, we can consider the impact of our model selection and experiment with regularized regressions and different linear models to better approximate the relationship between student behavior and performance.

Bibliography

[1] Karen Brennan and Mitchel Resnick. New frameworks for studying and assessing the development of computational thinking.

[2] Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. Studying learning in the worldwide classroom: Research into edx's first mooc. *Research & Practice in Assessment*, 8, 2013.

[3] Samantha R Cook, Andrew Gelman, and Donald B Rubin. Validation of software for bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3):675–692, 2006.

[4] Peadar Coyle and Benjamin Cook. Glm: Logistic regression. `http://http://docs.pymc.io/notebooks/GLM-logistic.html`, 2017.

[5] Nicholas Diana, Michael Eagle, John C Stamper, and Kenneth R Koedinger. Extracting measures of active learning and student self-regulated learning strategies from mooc data. In *EDM*, pages 583–584, 2016.

[6] Josh Gardner and Christopher Brooks. Student success prediction in moocs. *arXiv preprint arXiv:1711.06349*, 2017.

[7] Andrew Gelman. Everything i need to know about bayesian statistics, i learned in eight schools. `http://andrewgelman.com/2014/01/21/everything-need-know-bayesian-statistics-learned-eight-schools/`, 2014.

[8] Andrew Gelman, John B Carlin, Hal S Stern, and David B Dunson. *Bayesian data analysis*, volume 2. 2014.

[9] Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

[10] Christian Gütl, Rocael Hernández Rizzardini, Vanessa Chang, and Miguel Morales. Attrition in mooc: Lessons learned from drop-out students. In *International Workshop on Learning Technology for Education in Cloud*, pages 37–48. Springer, 2014.

[11] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[12] Kenneth R Koedinger, Jihee Kim, Julianna Zhuxin Jia, Elizabeth A McLaughlin, and Norman L Bier. Learning is not a spectator sport: Doing is better than watching for learning from a mooc. In *Proceedings of the second (2015) ACM conference on learning@ scale*, pages 111–120. ACM, 2015.

[13] Kenneth R Koedinger, Elizabeth A McLaughlin, Julianna Zhuxin Jia, and Norman L Bier. Is the doer effect a causal relationship?: how can we tell and why it's important. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 388–397. ACM, 2016.

[14] Michael J Lee and Andrew J Ko. Personifying programming tool feedback improves novice programmers' learning. In *Proceedings of the seventh international workshop on Computing education research*, pages 109–116. ACM, 2011.

[15] Youngju Lee and Jaeho Choi. A review of online course dropout research: Implications for practice and future research. *Educational Technology Research and Development*, 59(5):593–618, 2011.

[16] Cathy Sandeen. Integrating moocs into traditional higher education: The emerging âĂIJmooc 3.0âĂİ era. *Change: The magazine of higher learning*, 45(6):34–39, 2013.

[17] PyMC Development Team. A primer on bayesian methods for multilevel modeling.

[18] Phil Ventura and Bina Ramamurthy. Wanted: Cs1 students. no experience required. In *ACM SIGCSE Bulletin*, volume 36, pages 240–244. ACM, 2004.

[19] Thomas Wiecki. Why hierarchical models are awesome, tricky, and bayesian. `http://twiecki.github.io/blog/2017/02/08/bayesian-hierchical-non-centered/`, 2017.

[20] Brenda Cantwell Wilson and Sharon Shrock. Contributing to success in an introductory computer science course: A study of twelve factors. *SIGCSE Bull.*, 33(1):184–188, February 2001.

[21] Jeannette M Wing. Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881):3717–3725, 2008.

[22] A. Yan, M. J. Lee, and A. J. Ko. Predicting abandonment in online coding tutorials. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 191–199, Oct 2017.