Technology and Pedagogy: Using Big Data to Enhance Student Learning

Christopher Greg Brinton

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Electrical Engineering
Adviser: Mung Chiang

May 2016
Abstract

The “big data revolution” has penetrated many fields, from network monitoring to online retail. Education and learning are quickly becoming part of it, too, because today, course delivery platforms can collect unprecedented amounts of behavioral data about students as they interact with learning content online. This data includes, for example, each click made while watching a lecture video, while submitting an answer to a quiz question, or while posting a question on a discussion forum. The ability to capture this data presents novel opportunities to study the complex process by which learning occurs, and also raises interesting research questions around how behavioral data can be leveraged to improve the quality of each student’s learning experience, especially as online learning is scaled in size at the apparent expense of efficacy.

In this thesis, I detail three research thrusts we have undertaken in using big data to study learning and enhance pedagogy. First is Learning Data Analytics (LDA), in which we have developed new methods for representing student video-watching behaviors as compact sequences, extracted recurring patterns from these sequences and showed how certain ones are significantly correlated with performance, and used the results in the design of behavior-based, early detection algorithms for performance prediction. Second is Social Learning Networks (SLN), in which we have proposed a new model for social learning that combines the topical and structural aspects of discussions, used this model to determine the efficiency of existing discussions, and designed algorithms to encourage SLN formation around a more optimal state. Third is Integrated and Individualized Courses (IIC), in which we have developed two new learning technology systems – a student-facing, course delivery platform and an instructor-facing, analytics dashboard – that build models based on behavior, individualize the content delivered to students based on these models, and visualize certain components of the models to instructors. I will also discuss the extensions we
are exploring in terms of additional data capture, data analytics, algorithms, system
design, and user trials by deploying IIC in various learning scenarios.
Acknowledgements

A decade ago, I was a high school student that knew little more than a passion for mathematics and music. Fast forward ten years, and my intellectual interests have broadened, while my focus has gradually sharpened, to an area of research that truly suits me personally. The culmination of this is a doctorate from Princeton and a position as the Head of Research at a company that is almost identical to my thesis.

As my journey as a graduate student ends, I am proud of who I’ve become, and couldn’t be happier with where I’m headed. There are so many people who have influenced, inspired, and guided me over the years to whom I owe great thanks.

First are my amazing colleagues in the EDGE Lab over the past four years: Aakanksha, Aveek, Bharath, Carlee, Ehsan, Felix, Felix, Ioannis, Jaeyoon, Jessica, Jiasi, Liang, Mike, Sangtae, Shirley, Soumya, Swapna, Yixin, and Zhenming. I feel truly fortunate to have had the opportunity to collaborate with those of them that I worked with, and I will forever cherish the conversations, late nights and laughter that came along with sitting in C-319A. I could not have asked for a smarter group.

Next up is the incredible team that has been assembled at Zoomi, which started in 2013 with Bill, Mung, Ruediger, and Sangtae and has since evolved to include many full-timers: Bree, Caroline, Da, Jim, John, Nancy, Sergey, Tim, Weiyu, and Willy. They have kept the research, development, business, support, fundraising, and engine of innovation alive and thriving constantly at Zoomi, which has been a huge influence on my thesis. I am very excited to be officially joining them full time this summer.

I also must thank the outstanding professors who took the time to serve as my thesis readers and/or FPO committee members: Chee-Wei, Mung, Prateek, Sangtae, and Vince. Their comments throughout this process have been extremely valuable, and I hope to continue collaborating with each of them moving forward.

Of course, none of this would have been possible without Mung, my PhD adviser, 6P co-author, MOOC co-instructor, Zoomi co-founder, and all around mentor over
the past five years. I thank him for giving me the chance to work with him when I came to his office as an undergraduate student anxious for the opportunity to pursue graduate studies at Princeton. I will never forget the valuable lessons he has taught me about research, teaching, writing, entrepreneurship, professional development, and ambition for continuous self-improvement in general. His name always comes to mind whenever someone asks, “who is the smartest person you know?”

Before I even stepped foot on Princeton’s campus, I had made a couple of irreplaceable, lifelong friends as an undergraduate at The College of New Jersey: Dave, Kevin, Matt, Rachael, and Rob. I am so grateful that we have managed to stay close since we graduated in 2011, despite the fact that our professional lives no longer intersect. I thank them for having given me the nickname “Dr. Brinton” back in the day; it really did encourage me at different times towards translating it to reality.

There has always been something special about the close bond of my immediate family: Mom, Dad, Eric, and let’s not forget Baxter. Their unconditional love and faith in me has been so motivating in my graduate studies and everything else that has led to this point. It has given me the strength to push through my darkest times, to get up when I took my greatest falls. I would not have made it this far without their moral support, not to mention the support of my extended family, both current and soon-to-be: cousins, aunts, uncles, grandparents, the Hannans, and everyone else.

Last but certainly not least, I extend my deepest gratitude to Kate, the love of my life and soon-to-be-wife. She is beautiful, kind, compassionate, generous, patient, and – when it’s productive – brutally honest with me. I have felt very fortunate to have her at my side these past five years, and I look forward to all the memories we will make in the years to come, starting with bringing home our dog Sammy in June followed by our wedding in November.

A big thank you to each of these outstanding individuals.
To the Brintons, Hannans, and Lucariellos.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiv</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Learning Data Analytics</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Objectives and Contribution</td>
<td>11</td>
</tr>
<tr>
<td>2.1.1 Measuring Behavior and Performance</td>
<td>11</td>
</tr>
<tr>
<td>2.1.2 Behavioral Patterns</td>
<td>13</td>
</tr>
<tr>
<td>2.1.3 Performance Prediction</td>
<td>14</td>
</tr>
<tr>
<td>2.1.4 Summary of Contribution</td>
<td>15</td>
</tr>
<tr>
<td>2.2 Aggregate Video-Watching Quantities</td>
<td>16</td>
</tr>
<tr>
<td>2.2.1 My Two MOOCs</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2 Summary Quantities and Statistical Analysis</td>
<td>19</td>
</tr>
<tr>
<td>2.2.3 Key Messages</td>
<td>25</td>
</tr>
<tr>
<td>2.3 Video-Watching Sequences and Motifs</td>
<td>26</td>
</tr>
<tr>
<td>2.3.1 Processing Clickstream Events</td>
<td>26</td>
</tr>
<tr>
<td>2.3.2 Event-Type Sequence Specification</td>
<td>29</td>
</tr>
<tr>
<td>2.3.3 Motif Extraction</td>
<td>32</td>
</tr>
</tbody>
</table>
3.2 Decline Rate and Information Overload .................................................. 86
   3.2.1 Forum Dataset ................................................................. 86
   3.2.2 Analysis of Forum Activity Decline ........................................ 91
   3.2.3 Analyzing Information Overflow ......................................... 98
   3.2.4 Key Messages .................................................................. 101
3.3 Modeling SLN Efficiency ................................................................. 102
   3.3.1 Graph Modeling of SLN ($W$, $S$, and $D$) ............................ 102
   3.3.2 Utility Modeling of SLN ($B$ and $E$) ...................................... 103
   3.3.3 Optimizing SLN ................................................................. 105
   3.3.4 Computing the Observed Social Network ($\hat{W}$ and $\bar{w}_u$) .... 108
   3.3.5 Inferring Seeking and Disseminating Tendencies ($S$ and $D$) .... 110
   3.3.6 Solving for the Optimal Network ($W^*$ and $E^*$) ................. 115
3.4 Optimizing Interaction Structures .................................................... 117
   3.4.1 Datasets .................................................................... 117
   3.4.2 Extracting Topics and Q/A Tendencies .................................. 118
   3.4.3 Efficiency Evaluation ......................................................... 120
   3.4.4 Network Comparison ........................................................ 126
   3.4.5 Key Messages .................................................................. 131
3.5 Related Work ............................................................................. 132
   3.5.1 Studies on Improving MOOC Quality ................................. 132
   3.5.2 Forums for Online Education ............................................... 133
   3.5.3 Online Social Networks Outside MOOC .............................. 133
3.6 Extensions ................................................................................. 134
4 Integrated and Individualized Courses ........................................... 138
   4.1 AES in the Literature ............................................................. 140
      4.1.1 Navigation and Presentation Adaptation ......................... 140
      4.1.2 Adapting Learning Styles ............................................... 142
List of Tables

2.1 Basic information on the two datasets. The values in the bottom group of columns are the final numbers after the data has been filtered for analysis. ......................................................... 17

2.2 Identified intervals/sets with difference between CFA and non-CFA classes. The estimated $\hat{c}$, 95\% CI, and $p$-value are given for each. * denotes a $p$-value < 0.05, and ** is for < 0.01. ......................... 23

2.3 Tuned parameters for each of the algorithms. .............................................................. 53

2.4 Absolute performance metrics obtained from evaluating the algorithms over different subsets of the course data. Bold denotes the best achieved in each case. ................................................ ......................... 55

2.5 Summary of the tuned parameters and quality metrics obtained across the videos for each course. $b_v$ applies to the likelihood-based algorithms, while $C_v$ is for SM. The average (avg) and standard deviation (s.d.) are taken first over the 10 evaluation sets for each video, and then over all the videos. ......................................................... 69

2.6 $p$-values from applying WRS tests to the boxplots in Figure [2.13] . . . 72

3.1 List of course codes and their full names. All 78 were crawled and parsed. A * indicates a course that was used in the statistical analysis (73/78), and a △ indicates one that passed the Shapiro test (51/78, see Section [3.2.2]) ................................. 88
3.2 Regression parameters for forum activity models, with levels of significance indicated for each. ** indicates $p < 0.01$, * indicates $p < 0.05$, and . indicates $p < 0.1$. ................................. 97

3.3 Basic statistics of the four datasets, each corresponding to a different Coursera course session. The title, type (technical (T) / humanities (H)), start date (m/dd/yy), duration, and number of users, threads, and posts are given for each. ................................. 118

3.4 Summary of the topics extracted by Latent Dirichlet Allocation for each course, with $|K| = 10$. The topics are representative of likely discussions given the course context, and they tend to be non-overlapping. 119

4.1 Results of the four survey questions asking respondents to compare MIIC and OSFA. Results tended in favor of MIIC: one was significant at a confidence level of $\alpha = 0.05$ (*), and two were significant at $\alpha = 0.01$ (**)........................................ 173

4.2 Binary encoding of the version numbers for different segments in Figure 4.7.................................................. 175

4.3 Learning paths (omitting segments 1 and 8) among the 14 MIIC users who answered all the questions........................................ 176

4.4 $p$-values, 95% confidence interval (CI), and Hodges-Lehmann estimate (HLE) of the difference between the MIIC and OSFA groups for the distributions in Figure 4.10(a) and (b), respectively, using the Wilcoxon test........................................ 178
List of Figures

1.1 This emerging research area sits at the intersection of several disciplines and requires the application of several methodologies to improve learning quality. ................................. 5

2.1 The behavior considered here is video-watching behavior, and the performance is the score obtained on the quiz corresponding to a video. ................................. 12

2.2 Depiction of intervals of significance for continuous clickstream quantities, considering all videos and all users taking at least 20 quizzes. We can visually identify the difference between the CFA and non-CFA groups, particularly in (b) and (d). ........................................ 22

2.3 Illustration of a sequence of clicks $E_1$ to $E_4$ on a video, where the horizontal axis denotes the video length. This example will generate 5 events according to the model proposed here based on events and lengths. The length $l_j$ for the events that have this property (not pauses) are depicted above the diagram. ........................................ 27

2.4 Distribution of the lengths for four events across both ‘NI’ and ‘FMB’. For $P_1$ and $P_a$, this represents the time elapsed before the next event, and for $S_b$ and $S_f$, this is the distance of the skip. ........................................ 30

2.5 ECDFs of the number of sequences that each motif appears in, for both CFA and non-CFA. The supports are consistent between the CFA and non-CFA groups in each course. ................................. 35
2.6 ECDFs of the number of students that exhibit each motif, across both CFA and non-CFA sequences. In both courses, CFA sequences have a higher support for motifs across users. .......................... 36

2.7 ECDFs of the number of videos that each motif appears in, across both CFA and non-CFA sequences. In both courses, CFA sequences have a higher support for motifs across videos. .......................... 37

2.8 Representative sample of motifs identified for each course. Each motif is grouped by the dominant event it contains outside of P1. FS is the fraction of sequences over both CFA and non-CFA, while FS0 and FS1 are for the separate cases. ∆c is the estimated change in the probability of success (CFA) for every additional occurrence of the motif, and the p-value (p) is the significance of ∆c (a . indicates p ≤ 0.1, a * indicates p ≤ 0.05, and a ** indicates p ≤ 0.01). .......................... 39

2.9 Illustration of the key video-watching behaviors exhibited by each of the four motif groups in Figure 2.8. For brevity, we omit variations that exist between the individual motifs within the groups. In each case, the horizontal axis represents the video length, as in Figure 2.3. For (b) and (c), horizontal jumps represent skips with lengths relative to the distance, whereas the vertical jumps in (b) just indicate continuity. 40

2.10 Visualization of the design matrix X and target vector y for the SVM classification algorithms using aggregate quantities as features. ...... 48

2.11 Percent increases in prediction performance relative to the benchmark. VID-N and VID-A outperform the standard algorithms in each case for each metric; however, the incremental gain is the highest when considering VID-A in Ω^{12n}. .......................... 56
2.12 Plot of the fraction of local (repeat and direct) and non-local (backward and forward) transitions for each window size $w_v$, averaged over all UV Pairs for each position and video $v$. The fraction of non-local transitions is very low.

2.13 Boxplots of CFA prediction quality across both courses, considering accuracy and F1. Each datapoint is that measured on one of the videos considered. Qualitatively, we see that (i) the likelihood-based algorithms (DP, DT, and CT) outperform SR for both metrics, and (ii) the SVM-based (SM) algorithm also outperforms SR, but not as substantially.

2.14 Difference in quality (in percent) between DP and SR (i.e., DP minus SR) and between DP and SM (DP minus SM) across individual videos, for each course and each metric. A positive bar indicates that DP performed better than the algorithm, while a negative bar indicates it performed worse. DP outperforms both algorithms in the majority of cases, but it outperforms SR in more cases than it does SM.

3.1 The process of an instructor answering all student questions (left) does not scale with enrollment. In an SLN (right), the social component ideally allows the learning process to scale with the size of the network.

3.2 A MOOC discussion forum is a series of threads (left), with each thread having posts and comments (right).

3.3 Distributions of (a) student posts and (b) decline rates. Many students make only a few posts, and many courses have sharp decline rates.

3.4 Distribution of the number of students (left) and active students (right) per course, by category.
3.5 The decline of forum activities over time for 5 of the 73 courses, chosen randomly. In (a), the number of posts by day is plotted, and a regression line is added. In (b), Q-Q plots for these courses of the difference in count between two consecutive days are plotted, after removing 6% outliers. 

3.6 Distribution of thread lengths over the 73 courses used for statistical analysis. The average thread has about 5 posts, with a heavy-tail of a few threads receiving hundreds. 

3.7 Discussion thread length against the number of threads created around the same time, using a window size of 1. There is strong evidence that the creation of more threads in an interval of time is negative correlated with the attention each receives. 

3.8 There are six modules involved in determining the efficiency of an SLN. Section 3.3 describe our methods for obtaining the observed and ideal SLN, and Section 3.4 will describe the results of efficiency evaluation. 

3.9 Plate notation for Latent Dirichlet Allocation. 

3.10 Distributions of seeking \((s_{u,k})\) and disseminating \((d_{u,k})\) tendencies inferred for each dataset, for the two topics \(k\) with maximum support for each course (see Table 3.4). We can see that the \(d_{u,k}\) tend to be slightly larger than the \(s_{u,k}\), but that they are on the same order, implying there is typically sufficient disseminating tendency to match the questions posted on the topics.
3.11 Efficiency measures $\eta^g$ and $\eta^F$ as (a) the teaching benefit $\alpha_m$ and (b) the smoothing factor $\sigma$ are varied. In (a), even with $\alpha_m = 0$, the efficiencies are always below 1, highlighting the potential gains through optimization. Further, as $\alpha_m$ increases, the efficiency tends to drop, indicating that the networks are generally more formed around learning than teaching. In (b), we see that as $\sigma$ increases, the networks tend to become more efficient.

3.12 Plot of (a) the objective functions for the observed ($\hat{F}$) and optimized ($F^*$) networks, and (b) the global utilities for the observed ($\hat{g}$) and optimized ($g^*$) networks as the teaching benefit parameter $\alpha_m$ is varied. The ratios between the observed and optimized values correspond to the plots of $\eta^F$ and $\eta^g$ in Figure 3.11(a).

3.13 Plots of (a) the outgoing degree distributions and (b) the edge weight distributions for observed and optimal networks. For each dataset, we can see that optimization makes the distributions more uniform, with (a) less users having large outgoing degrees and (b) many additional connections established between users.

3.14 Distribution of the ratio of local to global utility, $r_u = l_u/g$, for the observed (obs) and optimal (opt) SLN in each course, with $\alpha_c = 0$. The median (med) and Jain’s Index (JI) of the plots are indicated in the caption, in the format: (med, JI). Given that the Jain’s Index values do not change substantially, we conclude that the optimization at least preserves the fairness of local utility.

3.15 Plot of the local utility $l_u$ for each user before (observed) versus after (optimized) optimization. The black line separates the regions of increased (above) and decreased (below) $l_u$. We can see that the majority of users have increased local utilities in each case.
3.16 Plot of local utility $l_u$ against (expected) outgoing degree $d_u$ for each of the datasets, before and after optimization. In each case, we can see that learners tend to obtain comparable $l_u$ for a lower $d_u$, especially in ml and comp where the separation is most profound.

4.1 IIC delivery is supported with two systems: a student-facing player and an instructor-facing dashboard. Under proper machine learning methodology, the behavioral measurements collected with the player lead to useful analytics and individualization.

4.2 Our AES design process. MIIC implements the subset of the options shown in bold.

4.3 A small example of using motifs, in addition to scores, to update the user model. Here, the UM has four dimensions: (i) revising on topic 1, (ii) scores on topic 1, (iii) revising on topic 2, and (iv) performance on topic 2. When a user watches a video containing these topics and takes a corresponding quiz, he/she revises on both topics and gets the quiz wrong. Her UM is changed accordingly (assuming quiz points and revisions have the same effect on the update for simplicity).

4.4 Diagrams to illustrate the definitions of learning path, segment-versions (seg-vers), and navigational and content adaptation.

4.5 Illustration of how the different learning modes were integrated into the tablet version of MIIC. In (a), we show video and corresponding text, along with the ability to select text and add notes, perform external searches, and so on; in (b), an assessment is presented to the user; in (c), the video and text bookmark menu is selected, and a bookmark icon is on the page; and in (d), a note created by another user is selected on this user’s tablet.
4.6. Architecture description of the MIIC implementation for the user trials. In (a), the key components coded and established for the user device and backend server are shown. In (b), high-level device-server interaction logic is shown, which is used to store user data, update the UM, fetch the next segment, and render it on the device. 158

4.7. Course structure for the MIIC employed in our student trials. 167

4.8. Illustration of a page in two different segments for MIIC in the first trial. (a) is from segment 2, which shows a more in-depth treatment of the subject. (b) is from segment 3, which only uses basic algebra to explain the fundamentals. 169

4.9. Student rating of overall experience with MIIC, on a five point scale. Over 80% responded “good” or “excellent.” 172

4.10. Distribution of the (a) pages and (b) objects accessed by students in the OSFA and MIIC groups. The distribution for MIIC is shifted to the right, indicating higher engagement (with the exception of one outlier). 177
Chapter 1

Introduction

Distance learning has surged tremendously over the past decade, in conjunction with advances in information technology. An increasing number of online learning platforms have emerged, servicing K-12, higher education, tutoring, professional certification, corporate training, and other learning scenarios [21]. Today, entire courses and learning programs exist online, with the eventual goal of providing global access to world class instruction [84].

For all its benefits, the quality of online learning has been the target of criticism. In comparing online to traditional, brick-and-mortar (i.e., face-to-face) instruction, research has found, for example, lower engagement and knowledge transfer to learners in higher education [102] and corporate training [66] courses, among other consequences. These poorer outcomes have been attributed to factors such as the asynchronous nature of interaction in online courses, which naturally places limitations on social learning between instructors and students and among students themselves [25].

One of the more recent innovations in this space that has stirred up controversy is the Massive Open Online Course (MOOC). MOOC providers such as Coursera, edX, and Udacity have offered courses reaching out to tens and even hundreds of
thousands of students within single sessions [21], demonstrating the potential of using the Internet to scale education to the masses. In fact, an estimated 35 million people signed up for at least one MOOC in 2015, an increase of 50% from the year before [92]. There are dozens of MOOC platforms today, differing to varying degrees in their business, operational, and/or certification models. But common across all of them is a salient observation that higher enrollments tend to be associated with higher dropoff rates. In fact, it is typical to see less than 10% of the students enrolled in a MOOC actually receiving a certificate of completion at the end of the course [26].

In addition to the asynchrony of online learning in general, many of the challenges with MOOC stem from its heterogeneity. The open nature of these courses attracts a diverse set of students with a range of motivations for why they enrolled in a course in the first place [115, 68], only a fraction of them identifying completion as one of their goals. Even among those students targeting completion, educational backgrounds are typically more diverse than what is seen in traditional classrooms [25]. This adds additional difficulty to the prospect of scaling the efficacy of traditional teaching methods with the size of the student body.

How can online learning be scaled up without hurting quality? This fundamental question is the primary motivator of this thesis. It has led to several, interrelated paths investigating the development of methods to make it possible. For one, are there certain types of information, or analytics, that would help instructors detect which students are most in need of assistance before they drop out? Also, can students possessing heterogeneous sets of knowledge be discovered and encouraged to collaborate on solving problems? Could it be possible, even, to automate the process of individualizing learning based on a student’s observed behavior?
Big Data on Student Learning

Investigating the research questions posed above is made possible in large part by the plethora of data that today’s online learning platforms can collect. The backend infrastructure driving these platforms is typically built to capture detailed measurements on students as they interact with the different forms of learning integrated into the courses.

For example, consider the standard three learning modes that MOOC platforms offer to students: video lectures, assessments (e.g., in-video quizzes, homework assignments, and exams), and social networking (usually through discussion forums) [21]. For video content, individual clickstream events are captured, where a click event is generated and stored each time a student interacts with a video which specifies the particular action (e.g., pause, ratechange, skip), position, and time at which it occurred. For assessments, the specific responses to individual questions are recorded. For the discussion forums, all threads, posts, comments, and votes are stored too.

This data brings substantial opportunities to study the process of student learning, and to design systems for improving learning quality. Indeed, after a few years and thousands of course offerings through these platforms, researchers have begun to take data mining approaches to studying student behavior (e.g., [65, 94, 73]). This has, in turn, motivated the design of mechanisms for outcome improvements. Notable examples including algorithms (i) for early detections of students with high likelihoods of dropping out of the course before its completion [112, 94] and of performing poorly on quizzes/exams [77], (ii) for recommendations of discussion participation [110] and of certain peer grading allocations [85], and (iii) for individualization of the content delivered to each student [74, 29].
Research Focus and Approach

Our research has focused on developing methods and systems to improve the quality of student learning. We have analyzed the types of behavioral data described above, and leveraged the resultant models in the development of technologies that perform two main functions: (i) generating intelligent analytics about learning behavior, to provide instructors with insights into how they and other students can provide better assistance to their students, and (ii) intervening into the course delivery process directly, through automated individualization of the learning path each student takes through the course. These data-driven technologies aim to help students, to help instructors help their students, and to help students help each other.

In particular, we have pursued the following three-pronged research approach:

(1) Massive Online (MOOC) Teaching. “Big” behavioral data has not been made open access for researchers. By co-instructing two MOOCs on Coursera – Networks Illustrated: Principles Without Calculus (‘NI’) [2] and Networks: Friends, Money, and Bytes (‘FMB’) [1] – we have been able to obtain learning data across hundreds of thousands of our own students.

(2) Data-Driven Learning Science. Using these datasets and others, our research projects have taken three thrusts:

- Learning Data Analytics (LDA) [22, 19] involves developing methodology to extract patterns from student behavioral data, and using the insights to design algorithms for learning analytics and individualization.

- Social Learning Networks (SLN) [25, 20, 21] consists of modeling the social networks emerging from peer-based learning, and designing algorithms to recommend better social interactions.
Learning Research: Principle Components

This thesis is part of a larger effort in learning quality enhancement, which has emerged as a field of research in the past few years [21]. This is an interdisciplinary area, involving engineering (electrical, systems, operations research), computer science (data mining, machine learning, algorithms), and education/pedagogy. It calls for the use of real-world data and experiences from teaching in the development of models and prototypes, and in turn incorporating these models and algorithms into...
systems deployed to various learning scenarios. The various components involved are summarized in Figure 1.1.

Glancing through the bodies of work in this area, a few principle components that cut across the field can be identified. These “intellectual themes” will define the next generation of learning technologies and motivate explorations moving forward: Recommendation, Prediction, Incentivization, Visualization, Integration, and Individualization.

**Recommendation**

Systems for recommending users to items have long been studied in computer science [70]. Online learning today calls for recommender schemes that address new challenges [49]. These include recommendations e.g., of specific content modules for a student to visit, of certain courses for a student to take [21], of communication between students that will lead to more effective social learning communities [20], and of certain behavioral patterns for a student to emulate [19]. These can be provided to students directly or fed to an instructor as a form of analytics. Data collected about interaction with different forms of content is important for building models of students and models of learning items to make such recommendations.

**Prediction**

Predictive modeling uses machine learning methods to predict outcomes from existing observations [79]. It is another key component to future learning technologies, with predictions e.g., of assessment, quiz, or exam scores [22], of engagement or dropoff locations [94], and of eventual course outcomes (like future job or task performance) to serve as useful instructor analytics. These analytics can lead to identification of specific students to help and specific content modules that may need modifications.
Prediction is also an important step in recommender systems that rely on intelligent estimates of how users would rate items.

Incentivization

An incentive is something that motivates an individual to perform an action, commonly modeled through utility theory in economics. In learning scenarios, this is encouragement e.g., to participate in discussions [54], to focus on lectures and study for exams, and to complete assignments. Given the poor outcomes observed in online courses, it is important to design incentive mechanisms beyond the prospect of high grades to fit specific scenarios properly. Examples that have demonstrated success in practice include reputation scores on Stackoverflow [8], badges for participation in MOOC forums [9], and gamification in learning [62].

Visualization

With large course enrollments and a plethora of data captured about each student, deciding on the right set of visuals and the right granularity of information to show instructors is challenging [24]. Developing effective dashboards for these scenarios requires close collaboration between UI/UX designers, data scientists, software engineers, and educators. These dashboards can visualize e.g., student learning paths [26], performance and engagement distributions [72], social learning networks [20], and the outputs of the prediction and recommendation components discussed above. The efficacy of these visuals can be quantified through usability testing and A/B testing comparing scenarios with and without them.

Integration

A course may have multiple modes of learning available for students, even within specific content modules. Integrating them together in a single application page gives
students opportunity for cognitive reinforcement from multiple perspectives and more opportunities to engage \[91\]. At the same time, it is important to decide on a proper UI so that the experience does not appear cluttered and overloaded. The same is true for visualizations in an instructor’s dashboard as well.

**Individualization**

Individualized instruction refers to tailoring the learning experience to meet the needs of each specific student \[74\]. A difficult process for teachers in traditional classrooms, it becomes even more challenging in settings like MOOC where there are much smaller teacher-to-student ratios, asynchronous interactions, and more heterogeneity to begin with. Automated personalization has traditionally relied on quiz responses, but that is only one component of what teachers use to differentiate learning effectively. A system achieving true individualization must be based on behavior, and must have a fine granularity of output \[26\]. The integration of various learning modes, and the machine-learning methods of recommendation and prediction are the first steps towards achieving this goal \[21\].

**Our Research Contribution**

The research thrusts we have undertaken pertain mostly to the prediction, recommendation, integration, and individualization components outlined above. These themes will be pointed out at different parts of this thesis. Before diving in to the core chapters, I will point out the key contributions that are made in them.

First, our work on Learning Data Analytics (Chapter 2) develops new methods for representing student video-watching behavior as compact sequences, and extracts recurring behavioral motifs from the clickstreams encoded in these representations. Additionally, it investigates the relationship between student behavior (on videos) and performance (on corresponding quizzes), and used the resulting insights to design
prediction algorithms for whether a student will succeed or fail on a particular quiz. These algorithms demonstrate an early detection capability, with quality improvement over standard baselines being particularly pronounced early in a course, underscoring their utility as instructor analytics.

In Social Learning Networks (Chapter 3), we have investigated the dynamics of user discussions, identifying course factors that are commonly associated with a decline in participation over time. Moreover, we have designed methodology for quantifying the efficiency of an SLN, by comparing observed network utility to a benchmark of maximum utility attainable through optimization. Using our framework, we have found that MOOC forums today tend to have rather low efficiency, calling for methods to shift SLN structures towards those that we obtain which will better benefit all users.

Finally, we created two novel learning technology systems – a student-facing, course delivery player and an instructor-facing, analytics dashboard – to support the delivery of a new type of course called an Integrated and Individualized Course (Chapter 4). IIC is defined by the abilities to integrate various modes of learning and to perform fine-granular, behavior-based individualization in delivering courses to students, with algorithms for user modeling based on machine-learning methods designed for LDA and SLN. User trials with our course-delivery player have demonstrated promising results in improving two learning outcomes – user response and engagement – with IIC thus far.
Chapter 2

Learning Data Analytics

Lectures are often the most popular mode of learning for students. This is true for courses delivered online as well as those in traditional classrooms. Compared with other modes like discussion forums and homework assignments, though, student behavior during lectures has received relatively little research attention to-date [65]. After a few years and thousands of course offerings through online learning platforms, video-watching behavioral data has been captured and is ready to be analyzed.

This focus of this chapter is some of our work in Learning Data Analytics (LDA) for student video-watching behavior [22] [19]. Broadly speaking, in these works, we studied the following main question: How is a student’s behavior related to his/her performance, and how can behavior be used to predict performance?

Developing an understanding of this has implications not only to theories about how humans process information, but also to systems aiming to improve student learning experiences. For one, early detection performance prediction systems – which are usually driven by past performance history – can be augmented with behavioral signals that were identified as being correlated with low or high student performance [77], as we have done recently for corporate training courses [40]. Additionally, algorithms for updating user models in individualization can be expanded to include
behavioral signals in making determinations as to the most suitable path of learning for each student to take, which we have implemented in our course-delivery platform recently as well. Furthermore, these relationships could be provided to course instructors directly, through visualization of extended learning and content analytics. The behavioral signals could give instructors insight into which parts and/or types of their content are causing confusion.

2.1 Objectives and Contribution

In this chapter, I will detail our investigation of the following specific subquestions of the main question posed above:

Q1. How can student behavior and performance be represented and quantified for analysis?

Q2. What patterns exist in a student’s learning behavior, and which of these are correlated with performance?

Q3. Can we use student behavior to predict their future performance better than we can without it?

2.1.1 Measuring Behavior and Performance

There are many ways that behavior and performance can be defined, depending in part on the specific course and dataset in question. In our work, we employed two datasets coming from two different MOOCs we instructed on Coursera, which will be described in Section 2.2.1. For these datasets, behavior and performance are measured as follows.
Figure 2.1: The behavior considered here is video-watching behavior, and the performance is the score obtained on the quiz corresponding to a video.

**Behavior**

We focus on the behavior students exhibited while watching lecture videos. This is the dominant mode of instruction that we provided in our courses, and is where users spend the majority of their time on MOOC platforms \([65, 73]\). These behaviors are captured through clickstream logs.

**Performance**

The videos in our courses are equipped with in-video quiz questions, which are short multiple-choice exercises that we designed to test a student’s knowledge recall of the content in the video before he/she proceeds. Our measures of performance are the scores that students obtain on their first attempts at these quizzes, \(i.e.,\) whether they are Correct on First Attempt (CFA) or not (non-CFA). We use the in-video questions because they serve as immediate feedback of the knowledge a student gained from the behavior they exhibited in the video, thereby reducing the effect of confounding factors (\(e.g.,\) reviewing other materials, varying information retention based on a student’s innate cognitive ability). The first attempt has also been selected in other
works as an objective measure of performance (in e.g., [101]). In general, slip and guess probabilities [113] could be inferred from subsequent quiz attempts; in our datasets, however, less than 9% of submissions have more than one valid attempt registered.

The goal, then, is to relate video-watching behavior to in-video quiz performance, as depicted in Figure 2.1. After filtering (see Section 2.3), the datasets for our two courses contain roughly 315K (i.e., 315 thousand) and 416K clickstream events corresponding to 26K and 36K first-attempt quiz submissions.

### 2.1.2 Behavioral Patterns

A few methods for representing student video-watching clickstreams have been proposed previously. Some have taken a higher-level approach and computed aggregate, summary quantities of the behaviors (e.g., fraction of video completed, duration of pause) [73], others have looked at the most frequently visited video positions [65], and others yet have searched for sequences of events in the clickstreams (e.g., play, then skip, then pause) [94]. We develop frameworks for representing clickstreams as sequences that encapsulate the important aspects of these approaches: the events, the positions in the video that a student visited, the duration or length of time between the events and positions, and/or aggregate quantities.

In Section 2.2, we will describe our datasets, and present an analysis of video-watching behavior in terms of aggregate quantities. In doing so, we will also see how certain watching characteristics – in particular, certain intervals of values taken by the features – are indicative of whether a user is more likely to be Correct on First Attempt (CFA) or not at answering a question. Then, in Section 2.3 we will develop an event-based framework to represent clickstreams, which captures event types and their lengths. Leveraging this framework, we will identify video-watching motifs, i.e., sub-sequences of student behavior that occur significantly often. These motifs by
themselves are informative of recurring behaviors, and additionally, we were able to correlate the occurrence of certain motifs with a change in the likelihood of CFA through mixed-effects modeling. For example, we found that a series of behaviors are indicative of students reflecting on material, and are associated with an increase in the chance of CFA in one of the courses. As another example, we identified motifs that are consistent with rapid-paced skimming through the material, and reveal that these are associated with a decrease in the chance of CFA in both of the courses.

For these patterns, the identified positive and negative correlations with CFA are particularly helpful, because for many of them, either case is conceivable. For one, skimming could intuitively be a sign of a student either correctly or incorrectly perceiving familiarity with the material; our results indicate the latter tends to occur more often. Also, we find that incorporating the lengths in addition to the events is important to these findings, because extracting motifs from sequences of events alone does not reveal these insights.

2.1.3 Performance Prediction

Motivated by existence of correlations between behavior and performance, we then developed predictors of knowledge gained based on the actions a student makes while watching a video. Enhancing student performance prediction (specifically, CFA prediction) is an important area of research in its own right, because such methods can improve systems for early detection of e.g., struggling/advanced students and easy/difficult material [26, 77]. The higher the quality of behavior-based prediction, the stronger the association between video-watching behavior and quiz performance, too.

We performed two studies of behavior-based prediction. First, in Section 2.4, we will discuss our design of a scheme that estimates CFA probabilities from the aggregate quantities and uses them as learning features for prediction. For comparison,
we will also present some standard algorithms that have been employed for CFA prediction using only performance data. In evaluating these methods, we will show that our scheme consistently outperforms the standard ones in different scenarios, and that the incremental gain is particularly high early in the course when there is little information about each learner. Importantly, this highlights the “early detection” capability of clickstream data.

The second study in Section 2.5 builds on this early detection finding, using several models on the sequences of positions that a learner visits in a video. In particular, we considered CFA prediction on a per-video basis, in order to quantify the benefit obtained by the positions in each individual video and to investigate the application of earliest detection. In evaluating four different algorithms, we will show that maximum likelihood-based algorithms obtain significant improvements in prediction when compared to a baseline that does not use click information, and that an SVM-based algorithm also obtains improvements over the baseline, though not as substantially. This again underscores the ability to relate clicks to knowledge gained, i.e., that video-watching behavior is related to quiz performance, and shows that behavioral information is useful in situations where multiple videos are not be available, like in short courses or for detection very early on in a course. Further, since the likelihood-based algorithms are directly based on student behaviors (as opposed to the SVM algorithm which learns a more complex function on top of the behaviors), they can generate analytics about content that are interpretable to the instructor.

2.1.4 Summary of Contribution

Compared with other work (Section 2.6), this chapter makes the following main contributions to the field of Learning Data Analytics:
• It develops two new frameworks for representing student video-watching behavior as sequences, and presents a set of aggregate quantities that can be used to summarize behavior further.

• It extracts recurring motifs of student video-watching behavior using motif identification schemes, and associates these fundamental patterns as well as more aggregate quantities with quiz performance.

• It demonstrates that video-watching behavior can be used to enhance student performance prediction on a per-video basis, e.g., for earliest detection, and that it can make substantial improvements over traditional quiz-based prediction.

2.2 Aggregate Video-Watching Quantities

In building up to the specification of aggregate quantities and their correlations, we will first describe the two courses and datasets used in this chapter.

2.2.1 My Two MOOCs

Our datasets come from two different courses that we have instructed on Coursera: Networks: Friends, Money, and Bytes (‘FMB’) [1] and Networks Illustrated: Principles Without Calculus (‘NI’) [2]. Each of these courses teach networking topics, but ‘FMB’ delves into the mathematical specifics behind the topics, whereas ‘NI’ is meant as an introduction to the subject (see Chapter [3] for more details on the differences in the content).

Course Formats

The course formats are summarized in Table [2.1]. Each is made up of a series of lectures, which are in turn comprised of a set of videos. ‘FMB’ is a longer course,
with 20 lectures, whereas ‘NI’ only has 6 lectures. ‘NI’ had more, shorter-length videos, with a total of 115 videos and an average (avg.) length of 5.4 min per video, whereas ‘FMB’ has less, longer-length videos, with 93 total videos and an average length of 16.9 min per video.

As mentioned in Section 2.1.1, each course included in-video quizzes at the end of the videos. These quizzes were designed to test a student’s recall of the information discussed in the video. Besides in-video quizzes, our courses had two other machine-graded assessments: a midterm and a final exam. The exams in our MOOCs are much less suitable for quality evaluation than are the in-video quizzes, though, because (i) only a small fraction of students actually took them (less than 5% in each case), and (ii) there were only a small proportion of correct submissions since the exam questions were designed to be much more difficult than the in-video quizzes.

Each in-video quiz is a multiple choice question, in radio-response format, with 4-5 possible answer choices. For ‘FMB’, there was one question at the end of each video, whereas for ‘NI’, each of the 69 questions was associated with anywhere from 1-4 videos. In mapping videos to quizzes, we refer to “video n” as the contiguous set of videos occurring after question n – 1 and before question n.
User-Video (UV) Pairs

We obtained two types of data from Coursera for each of the courses. First is the video-watching clickstreams, which log user interaction with the video player. In particular, these logs contain one entry for each time an event – play, pause, rate change, or seek – was fired, specifying the user and video IDs, type, playback position, playback speed, and UNIX time of the event. Second is the in-video quiz submissions, which contains the user ID, quiz ID, time of submission, and answer selected for each attempt.

From these two types, we extracted User-Video (UV) Pairs from the data, with two sets of information for video and quiz \( n \):

- **Video-watching trajectory**: The set of video-watching clickstream logs (events) for the user in video \( n \).

- **CFA result**: Whether the user was Correct on First Attempt (CFA) or not (non-CFA) for quiz \( n \).

It is worth mentioning that, by focusing on those UV Pairs for which the student actually submitted an answer to the question, we are also focusing on a sample of the students for which obtaining high quiz performance was likely a consistent motivation.

In total, for ‘FMB’ there were 123K (i.e., 123 thousand) UV Pairs with 566K click events, and for ‘NI’ these numbers were 149K and 882K, respectively. Then, after removing any UV Pair that had at least one null, stall or error contained in its video-watching trajectory, we obtain the totals given in Table 2.1. The average CFA score across the UV Pairs are also shown here: 0.663 for ‘FMB’ (standard deviation (s.d.) = 0.473), and 0.750 for ‘NI’ (s.d. = 0.433).

The number of observed UV Pairs is large, but also rather sparse if we consider the numbers that we would have to work with if all the students answered all the questions in each course. In particular, only 7.6% and 19.7% of the possible pairs are
present in ‘FMB’ and ‘NI,’ respectively. The large number of unanswered questions is one of the challenges to MOOC performance prediction in the first place. We will see in Sections 2.4 and 2.5 how video-watching behavior improves the quality CFA prediction in the presence of this sparsity.

2.2.2 Summary Quantities and Statistical Analysis

For each UV Pairs, we extracted the following nine aggregate, summary quantities to describe behavior:

1. **Fraction spent** (fracSpent): The fraction of (real) time the user spent playing the video, relative to its length.

2. **Fraction completed** (fracComp): The percentage of the video that the user played, not counting repeated play position intervals; hence, it must be between 0 and 1.

3. **Fraction played** (fracPlayed): The amount of the video that the user played, with repetition, relative to its length.

4. **Number of pauses** (numPaused): The number of times the user paused the video.

5. **Fraction paused** (fracPaused): The fraction of time the user spent paused on the video, relative to its length.

6. **Average playback rate** (avgPBR): The time-average of the playback rates selected by the user. The player on Coursera allows rates between 0.75x and 2.0x the default speed.

---

1 Subsequent attempts are even more sparse. If we consider those attempts made by a student on a quiz occurring (i) after the first one, (ii) at least three seconds after the previous one (so it is not obviously a random guess), and (iii) at most two minutes after the previous one (so the student did not obviously move away from the question to look for the answer), only 8.2% and 5.5% of the user-quiz pairs in ‘FMB’ and ‘NI’ have more than one attempt.

2 See Section 2.3.1 for clickstream processing and denoising procedures.

3 Some of these are similar to those defined in [73].
7. **Standard deviation of playback rate** \( (\text{stdPBR}) \): The standard deviation of the playback rates selected over time.

8. **Number of rewinds** \( (\text{numRWs}) \): The number of times the user skipped backward in the video.

9. **Number of fast forwards** \( (\text{numFFs}) \): The number of times the user skipped forward in the video.

These quantities are not independent of each other, but each gives different insight into user behavior.

**Statistical Analysis Methods**

Two types of analysis were performed for each quantity, in order to see how each one varies between CFA and non-CFA instances. First, we examined where the probability density lies, and determined whether there is an overall difference in the distributions for the CFA and non-CFA groups. Since Shapiro-Wilk tests detected significant departures from normality for the distributions, we ran the non-parametric Wilcoxon Rank Sum (WRS) test \[93\] for the null hypothesis that there is no difference between the classes overall. We will report the \( p \)-value \( (p_W) \) from this test, and when it is low enough (say, below 0.05), we can reject the null hypothesis and assume the difference is significant.

Second, we considered whether there are certain intervals or sets of values that indicate a higher likelihood of a UV Pair falling in one of the groups. We identified the potential intervals by visually analyzing the probability density of the two groups; for continuous quantities 1-3 and 5 shown in Figure 2.2, we used Gaussian Kernel Density Estimation \[79\] with a bandwidth parameter \( \eta \) stated in each case. For each of the intervals, we ran a two-sample test for proportions \[93\] for the null hypothesis that there is no difference between the fraction of CFA and non-CFA samples occurring there, relative to the totals for each group. If the \( p \)-value from this test is low, then
there is a large enough difference between the fractions and a large enough sample size in the interval to assume that the CFA probability estimate $\hat{c}$ is significant.\footnote{In other words, a significant $p$-value tells us we can trust $\hat{c}$. Then, if $\hat{c}$ is above 0.5, $\hat{c} - 0.5$ tells us how much more likely CFA is in that interval; if below 0.5, then 0.5$ - \hat{c}$ tells us how much more likely non-CFA is.} For these cases, we report the $p$-value, $\hat{c}$, and a 95% Confidence Interval (CI) around $\hat{c}$, all of which are tabulated in Table 2.2.\footnote{It is worth mentioning that even within specific intervals, the specific user and video could be confounding factors influencing the score. This problem could be rectified in future work through mixed-effects modeling, as is done for the motifs in Section 2.3.}

For this analysis, we focused on the ‘FMB’ dataset; the results would be qualitatively similar for ‘NI’. Within ‘FMB’, we considered all videos, but only the active users who answered at least 20 questions, for which there are roughly 9.2K CFA and 4.7K and non-CFA samples.

**Playing Behavior**

This corresponds to Quantities 1 to 3.

**fracSpent:** Much of the density is in $[0.9, 1.1]$ (40% of CFA, 42% of non-CFA). The mean for CFA is 0.82 (standard deviation (s.d.) = 0.36), compared to a mean of 0.78 (s.d. = 0.39) for non-CFA. This indicates that, as expected, a user submitting a correct answer tends to have spent more time with the video. The difference between the distributions is significant ($p_W = 4.7E-8$). As shown in Figure 2.2(a), we identified three intervals of interest: $[0, 0.54]$ for which there is more non-CFA density, and $[0.54, 0.90]$ and $[1.1, 2.0]$ with more CFA, giving $\hat{c}$ of 0.45, 0.52, and 0.51. Moving from the first interval to the second, the chance of success increases by about 7%.

**fracComp:** Here, much density lies in $[0.95, 1]$ (57% of both classes). The mean for CFA is 0.76 (s.d. = 0.35), as opposed to 0.74 (s.d. = 0.37) for non-CFA. The difference between the distributions not significant ($p_W = 0.443$). Still, we identified two intervals of significance (Figure 2.2(b)): $[0, 0.13]$, where there is more non-CFA
density, and [0.76, 0.95], with more CFA. They give $\hat{c}$ of 0.47 and 0.53, or a 6% increase moving from the first to the second.

fracPlayed: For this, much of the density is in [0.9, 1.1] (64% of CFA, 62% of non-CFA). The mean for CFA is 0.91 (s.d. = 0.36), as opposed to 0.85 (s.d. = 0.39) for non-CFA. Similar to fraction spent, this indicates that a student submitting CFA tends to watch more of the video, and the difference between the distributions is significant ($p_W = 4.2E-24$). We found two intervals (Figure 2.2(c)): [0, 0.8], with
Table 2.2: Identified intervals/sets with difference between CFA and non-CFA classes. The estimated $\hat{c}$, 95% CI, and $p$-value are given for each. * denotes a $p$-value $< 0.05$, and ** is for $< 0.01$.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Interval/Set</th>
<th>$\hat{c}$ (%)</th>
<th>95% CI</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>fracSpent</td>
<td>$[0, 0.54]$</td>
<td>45.2</td>
<td>(0.438, 0.466)</td>
<td>$5.27\times10^{-12}$**</td>
</tr>
<tr>
<td></td>
<td>$[0.54, 0.90]$</td>
<td>51.9</td>
<td>(0.504, 0.534)</td>
<td>0.017*</td>
</tr>
<tr>
<td></td>
<td>$[1.1, 2.0]$</td>
<td>51.1</td>
<td>(0.500, 0.521)</td>
<td>0.049*</td>
</tr>
<tr>
<td>fracComp</td>
<td>$[0, 0.13]$</td>
<td>47.1</td>
<td>(0.459, 0.483)</td>
<td>$1.63\times10^{-6}$**</td>
</tr>
<tr>
<td></td>
<td>$[0.76, 0.95]$</td>
<td>52.7</td>
<td>(0.517, 0.537)</td>
<td>$3.59\times10^{-7}$**</td>
</tr>
<tr>
<td>fracPlayed</td>
<td>$[0, 0.80]$</td>
<td>43.7</td>
<td>(0.422, 0.451)</td>
<td>$2.2\times10^{-16}$**</td>
</tr>
<tr>
<td></td>
<td>$[1.05, 1.65]$</td>
<td>55.4</td>
<td>(0.541, 0.567)</td>
<td>$6.9\times10^{-15}$**</td>
</tr>
<tr>
<td>numPaused</td>
<td>${0, 1}$</td>
<td>42.0</td>
<td>(0.403, 0.437)</td>
<td>$2.2\times10^{-16}$**</td>
</tr>
<tr>
<td></td>
<td>${2, ..., 10}$</td>
<td>56.7</td>
<td>(0.550, 0.584)</td>
<td>$1.35\times10^{-14}$**</td>
</tr>
<tr>
<td>fracPaused</td>
<td>$[0.01, 0.082]$</td>
<td>54.0</td>
<td>(0.527, 0.553)</td>
<td>$3.97\times10^{-9}$**</td>
</tr>
<tr>
<td></td>
<td>$[0.082, 0.25]$</td>
<td>46.4</td>
<td>(0.450, 0.477)</td>
<td>$6.06\times10^{-8}$**</td>
</tr>
<tr>
<td></td>
<td>$[0.28, 0.356]$</td>
<td>50.7</td>
<td>(0.501, 0.513)</td>
<td>0.022*</td>
</tr>
<tr>
<td>avgPBR</td>
<td>${1}$</td>
<td>48.1</td>
<td>(0.463, 0.498)</td>
<td>0.032*</td>
</tr>
<tr>
<td></td>
<td>$\mathbb{R}_{\geq 0} \setminus {1}$</td>
<td>52.9</td>
<td>(0.512, 0.546)</td>
<td>$8.2\times10^{-4}$**</td>
</tr>
<tr>
<td>stdPBR</td>
<td>${0}$</td>
<td>46.3</td>
<td>(0.448, 0.477)</td>
<td>$1.11\times10^{-6}$**</td>
</tr>
<tr>
<td></td>
<td>$\mathbb{R}_{\geq 0} \setminus {0}$</td>
<td>53.0</td>
<td>(0.523, 0.552)</td>
<td>$1.11\times10^{-6}$**</td>
</tr>
<tr>
<td>numRW</td>
<td>${0}$</td>
<td>42.9</td>
<td>(0.413, 0.446)</td>
<td>$&lt; 2.2\times10^{-16}$**</td>
</tr>
<tr>
<td></td>
<td>${1, ..., 5}$</td>
<td>53.3</td>
<td>(0.519, 0.548)</td>
<td>$1.07\times10^{-5}$**</td>
</tr>
</tbody>
</table>

more non-CFA density, and $[1.05, 1.65]$ with more CFA, giving $\hat{c}$ of 0.44 and 0.55. This 11% difference is the largest out of the three playing behaviors.

**Pausing Behavior**

This corresponds to Quantities 4 and 5.

**numPauses**: Much of the density is in $\{0, 1\}$ (60% of CFA, 67% of non-CFA). The mean for CFA is 1.77 (s.d. = 2.05), as opposed to 1.48 (s.d. = 1.84) for non-CFA. This indicates that students who get questions correct tend to pause more (i.e., to reflect on the material), and the overall difference is significant ($p_W = 2.7\times10^{-16}$). We identified two sets of interest: $\{0, 1\}$ with more non-CFA density, and $\{2, ..., 10\}$ with
more CFA. They give $\hat{c}$ of 0.42 and 0.57, for a substantial change of 15% between the sets, the largest for the pausing behaviors.

$\text{fracPaused}$: Here, much density lies in $[0, 0.01]$ (32% of CFA, 35% of non-CFA). The mean of each class is roughly 0.13 (s.d. = 0.20), and there is no significant difference between the distributions ($p_W = 0.253$). Even so, we identify three intervals (Figure 2.2(d)): $[0.01, 0.082]$ and $[0.28, 0.356]$, where there is more non-CFA density, and $[0.082, 0.25]$, where there is more CFA, giving $\hat{c}$ of 0.54, 0.51, and 0.46.

**Playback Rate Behavior**

This is for Quantities 6 and 7.

$\text{avgPBR}$: Much density is at 1 (63% for non-CFA, 60% for CFA), indicating that many keep the default rate. Interestingly, even though the mean for both classes is roughly the same at 1.17 (s.d. = 0.28), the difference between them is significant ($p_W = 0.018$). We identified two sets: 1, with more non-CFA density, and $\mathbb{R}_{>0} \setminus 1$, with more CFA, giving $\hat{c}$ of 0.48 and 0.53.

$\text{stdPBR}$: For this quantity, much of the density is at 0 (79% for non-CFA, 75% for CFA), meaning that many hold the playback rate constant. The mean for non-CFA is 0.011 (s.d. = 0.043), while that for CFA is 0.015 (s.d. = 0.049). The difference between the distributions is significant ($p_W = 1.3E-7$), indicating that CFA tends to change the playback rate more. We identified two sets: 0, with more non-CFA density, and $\mathbb{R}_{>0}$, with more CFA density, giving $\hat{c}$ of 0.46 and 0.53.

**Skipping Behavior**

Finally, this is for Quantities 8 and 9.

$\text{numRWs}$: Here, much density is at 0 (78% for non-CFA, 73% for CFA). The mean for non-CFA is 0.46 (s.d. = 1.04), compared to 0.61 (s.d. = 1.21) for CFA. There is a significant difference between the distributions ($p_W = 3.5e-10$), indicating that CFA
tends to rewind more (i.e., revisit material). We consider two sets: 0 and \{1, ..., 5\}, with a higher concentration of non-CFA and CFA, respectively, giving \( \hat{c} \) of 0.43 and 0.53 for a reasonably large 10% difference.

\textbf{numFFs}: The density is largest at 0 (79% for both classes), and the means for both classes are roughly 0.42 (s.d. = 0.99). There is no significant difference between the classes \((p_W = 0.768)\), and we found no sets of interest. When considering skip back events as parts of revising sequences in Section 2.3, however, we will find significance in this behavior.

\section*{2.2.3 Key Messages}

We conclude from this analysis that satisfying at least one of the following characteristics is an indication that a user has a higher chance of CFA than not on a quiz:

- \textit{Playing behavior}: Playing more of the video than its length, spending more time on a video than its length, or completing more than 3/4ths of a video (but not its entirety).

- \textit{Pausing behavior}: Pausing more than once, or pausing either for a very short or very long time relative to the video length.

- \textit{Playback rate behavior}: Having an average playback rate different from the default speed, or varying the playback rate.

- \textit{Jumping behavior}: Rewinding at least once.

When displayed to an instructor as visualizations on an analytics dashboard, these can give him/her indication as to which characteristics serve as signals for success. Further, a user who is either new or struggling could take these as points of advice on how to potentially increase his/her chance of success. We will turn these indications to online performance prediction in Section 2.4.
2.3 Video-Watching Sequences and Motifs

The aggregate quantities in the previous section are intuitive ways of summarizing how a learner watched a video. On the other hand, they fail to capture the sequences in which events occur, as well as the specific positions in the videos where they are made. In this section, we move to developing methods for representing clickstreams as compact sequences, attempting to capture the important aspects of behaviors that occur repeatedly and exhibit differences between CFA and non-CFA.

2.3.1 Processing Clickstream Events

Formally, let $E_i$ denote the $i$th click event that occurs while a user is watching a video. We write $E_i = (e_i, p_i, t_i, s_i, r_i)$, where $e_i$ is the type of the $i$th click, $p_i$ is the video position of the player (in seconds) right after $E_i$ is fired, $t_i$ is the UNIX time (in sec) at which $E_i$ was fired, $s_i$ is the state of the video player – either playing or paused – after the click $E_i$ occurs, and $r_i$ is the playback rate (i.e., speed) of the video player resulting from this event. The logs are sequenced chronologically for a User-Video (UV) Pair, i.e., $t_1 < t_2 < \cdots$.

Based on the $E_i$ for a UV Pair, we define the following events (with some similarities but important differences from the quantities in Section 2.2):

**Play** (Pl): A play event begins at the time when a click event $E_i$ is made for which the state $s_i$ is playing, and lasts until the next click $E_{i+1}$. It occurs for a duration $d = t_{i+1} - t_i$ and has a length $l = p_{i+1} - p_i$.

**Pause** (Pa): A pause event is defined in the same way as a play event, except it is for which the state $s_i$ is paused, and does not have any length by definition.

**Skip back** (Sb): A skip back (i.e., rewind) event occurs when the type $e_i = \text{skip}$ and $p'_i > p_i$, where $p'_i$ is the position of the video player immediately before the skip.
playing, \( r > \)

\(< \text{skip}, p_2, t_2, \text{playing}, r >\)

\(< \text{skip}, p_4, t_4, \text{paused}, r >\)

\(< \text{pause}, p_3, t_3, \text{paused}, r >\)

\(< \text{play}, p_1, t_1, \text{playing}, r >\)

Figure 2.3: Illustration of a sequence of clicks \( E_1 \) to \( E_4 \) on a video, where the horizontal axis denotes the video length. This example will generate 5 events according to the model proposed here based on events and lengths. The length \( l_j \) for the events that have this property (not pauses) are depicted above the diagram.

If \( s_{i-1} = \text{playing} \), then \( p'_i = p_{i-1} + (t_i - t_{i-1}) \cdot r_{i-1} \); if \( s_{i-1} = \text{paused} \), then \( p'_i = p_{i-1} \).

The length of the skip is \( l = |p_i - p'_i| \), and there is no associated duration.

**Skip forward** (Sf): A skip forward (i.e., fast forward) event is defined as \( \text{Sb} \), except it captures the case where \( p_i > p'_i \).

**Ratechange fast** (Rf): This occurs when \( e_i = \text{ratechange} \) and the new rate \( r_i > 1.0 \).

There is no duration or length.

**Ratechange slow** (Rs): This occurs when \( e_i = \text{ratechange} \) and \( r_i < 1 \), again with no duration or length.

**Ratechange default** (Rd): This is when \( e_i = \text{ratechange} \) and \( r_i = 1 \), i.e., the user is returning to the default speed.

With these definitions, the sequence of events for a UV Pair becomes \((\hat{e}_1, \hat{e}_2, ...)\), for \( \hat{e}_j \in \mathcal{E} = \{\text{Pl}, \text{Pa}, \text{Sb}, ...\} \), \(|\mathcal{E}| = 8\). Each \( \hat{e}_j \) may have an associated duration parameter \( d_j \) and/or length parameter \( l_j \). Figure 2.3 shows a schematic for illustration; in this example, the clickstreams would generate: \( \text{Pl} \), with \( l_1 = (t_2 - t_1) \cdot r \) and \( d_1 = t_2 - t_1 \); \( \text{Sf} \), with \( l_2 = p_2 - p'_2 \); \( \text{Pl} \), with \( l_3 = p_3 - p_2 \) and \( d_3 = t_3 - t_2 \); \( \text{Pa} \), with \( d_4 = t_4 - t_3 \); and \( \text{Sb} \), with \( l_5 = p'_4 - p_4 \). Note that we are inserting \( \text{Pl} \) and \( \text{Pa} \) events in-between other events, to incorporate the state of the video player during those times. This critical

---

[6] On Coursera, the default player speed is 1.0. Users can vary this between 0.5 and 2.0, in increments of 0.25.
information is not captured through the raw events alone, and has been neglected in other work (e.g., in [94, 73]). In fact, only the Sf and Sb events are identical to aggregate quantities defined in Section 2.2.

De-Noising Clickstreams

It is important to remove noise in the video-watching trajectories associated with unintentional user behavior. We handle two cases of events separately.

**Combining repeat events.** First, we combine repeated, sequential events that occur within a short duration (5 seconds) of one another, since this pattern indicates that the user was adjusting to a final state. This is a common occurrence with forward (Sf) and backward (Sb) skips, where a user repeats the same action numerous times in a few seconds in seeking the final position; this should be treated as a single skip to the final location. Similarly, a series of rate change (Rf, Rs, or Rd) events may occur in close proximity, indicating that the user was in the process of adjusting the rate to the final value, which should also be treated as a single event. Formally, if there is a sequence of clicks $E_i, E_{i+1}, ..., E_{i+K}$ for which $e_i = e_{i+1} = \cdots = e_{i+K}$ and $t_{i+k+1} - t_{i+k} < 5 \ \forall k \in \{0, ..., K-1\}$, then we use $E'_i = \langle e_i, p_{i+K}, t_i, s_{i+K}, r_{i+K} \rangle$ in place of $E_i, E_{i+1}, ..., E_{i+K}$.

**Discounting invalid intervals.** Second, even though clickstream logs are the most detailed accounts of a student’s video-watching behavior that are available for online courses today, it is not possible to determine with complete certainty whether a student is actually watching and/or focused on the video for the duration of time in-between the occurrence of two events. Still, we can identify two situations in which a Pl or Pa event should not be inserted in-between $E_i$ and $E_{i+1}$ to capture the state of the video player. The first situation is if the duration $t_{i+1} - t_i$ is extremely long; in this case, the user was obviously engaging in some off-task behavior during this time. If $s_i = \text{paused}$, the threshold on the duration is set to 20 minutes (as in [104] for a
user’s inactivity on a website); if $s_i = \text{play}$, then the threshold is set to the length of the video. The second situation is if $E_i$ and $E_{i+1}$ occur on two different videos; here, there is no continuity as the user must have exited the first video and opened the second.

### 2.3.2 Event-Type Sequence Specification

An important part of the sequence specification is how to discretize the length $l_j$ and duration $d_j$ of the events. To this end, Figure 2.4(a) gives the boxplots of the event distributions from each course. $d_j$ for P1 and Pa is shown, and we depict $l_j$ for Sb and Sf, giving values that are at least 0.1 sec. Basic statistics of each distribution are also given in Figure 2.4(b); specifically, the three quartiles $Q_1$, $Q_2$, and $Q_3$ are shown as are the number of events for each distribution (Size) and the respective fractions.

#### Length Comparisons

We make three high-level observations in comparing the distributions. In each case, we employed a Wilcoxon Rank Sum (WRS) test \[93\] for the null hypothesis that there was no difference between the distributions for each dataset overall, and report the $p$-values from those tests.\[8\]

(i) ‘FMB’ has longer events: The distributions for each event (P1, Pa, Sb, and Sf) are shifted to the right for ‘FMB’ relative to those for ‘NI’, meaning that ‘FMB’ tends to have longer events. In each of the four cases, the $p$-values were highly significant ($\approx 0$). The fact that Pa is longer for ‘FMB’ is consistent with this subject material being more difficult.

(ii) Sf is longer than Sb: The distribution of Sf is shifted to the right relative to Sb for both ‘FMB’ and ‘NI’ ($p$-value < 1E-6). This indicates that when students skip

---

\[7\]By definition, quartiles separate data in increments of 25%.

\[8\]As in Section 2.2 we use the WRS test because Shapiro-Wilk tests detected significant departures from normality for each of the distributions.
Distributions of $d_j$ (Pl and Pa) and $l_j$ (Sb and Sf), in sec

- $10^{-1}$
- $10^0$
- $10^1$
- $10^2$
- $10^3$

Pl (FMB)
Pa (FMB)
Sb (FMB)
Sf (FMB)
Pl (NI)
Pa (NI)
Sb (NI)
Sf (NI)

(a) Boxplots of the distributions for each dataset.

(b) Tabulated statistics for the distributions.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Event</th>
<th>Size</th>
<th>Fraction</th>
<th>$Q_1$</th>
<th>$Q_2$</th>
<th>$Q_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>'FMB'</td>
<td>Pl</td>
<td>112.7K</td>
<td>53%</td>
<td>13.9</td>
<td>67.5</td>
<td>282.4</td>
</tr>
<tr>
<td></td>
<td>Pa</td>
<td>51.2K</td>
<td>24%</td>
<td>9.6</td>
<td>31.9</td>
<td>102.4</td>
</tr>
<tr>
<td></td>
<td>Sb</td>
<td>29.4K</td>
<td>14%</td>
<td>17.7</td>
<td>35.4</td>
<td>72.7</td>
</tr>
<tr>
<td></td>
<td>Sf</td>
<td>18.2K</td>
<td>8.6%</td>
<td>21.2</td>
<td>63.7</td>
<td>227.2</td>
</tr>
<tr>
<td>'NI'</td>
<td>Pl</td>
<td>103.5K</td>
<td>58%</td>
<td>12.0</td>
<td>71.0</td>
<td>262.6</td>
</tr>
<tr>
<td></td>
<td>Pa</td>
<td>46.4K</td>
<td>26%</td>
<td>4.5</td>
<td>19.3</td>
<td>58.8</td>
</tr>
<tr>
<td></td>
<td>Sb</td>
<td>17.8K</td>
<td>10%</td>
<td>12.9</td>
<td>26.2</td>
<td>54.7</td>
</tr>
<tr>
<td></td>
<td>Sf</td>
<td>10.7K</td>
<td>6.0%</td>
<td>9.6</td>
<td>28.4</td>
<td>81.7</td>
</tr>
</tbody>
</table>

Figure 2.4: Distribution of the lengths for four events across both ‘NI’ and ‘FMB’. For Pl and Pa, this represents the time elapsed before the next event, and for Sb and Sf, this is the distance of the skip.

forward, they tend to pass more material than they revise when skipping back. Sb also occurs more frequently than does Sf for both courses.

(iii) Pl is longer than Pa: The distributions for Pl and Pa in both datasets indicate that users tend to stay in the playing state longer than in the paused state ($p$-value $\approx 0$). This effect is stronger in the case of ‘NI’, which is again consistent with the fact that the ‘FMB’ material is more difficult.

**Sequence Specification**

Clearly, length $l_j$ and duration $d_j$ can vary substantially between events and datasets.

To account for this relative variation, we will use the four intervals in-between the
three quartiles for each event (given in Figure 2.4(b)) to discretize the lengths. We specify three cases:

(i) \( \hat{e}_j \in \{Sb, Sf\} \): When the event is a skip, we map it to \( \hat{e}_j q_j \), where \( q_j \in \{1, 2, 3, 4\} \) is chosen such that \( l_j \in (Q_{q_j-1}, Q_{q_j}) \), with \( Q_0 = 0 \) and \( Q_4 = \infty \). For example, suppose that event \( E_i \) is such that \( \hat{e}_j = Sb \) and \( l_j = 20 \) sec. In either course, this would be mapped to \( Sb2 \).

(ii) \( \hat{e}_j = Pa \): In this case, the mapping works the same as in the first case, except \( q_j \) is chosen based on \( d_j \) instead.

(iii) \( \hat{e}_j = Pl \): Two long duration play events could still have different qualitative interpretations. To account for this, when \( \hat{e}_j = Pl \), we map it to \( \hat{e}_j q_{j,1} \hat{e}_j q_{j,2} \cdots \hat{e}_j q_{j,K} \), where \( q_{j,k} \in \{1, 2, 3\} \) for \( k = 1, \ldots, K \) is chosen according to:

\[
q_{j,k} = \begin{cases} 
3, & d_j - \delta_{j,k} > Q_3 \\
\arg\min_{q_{j,K}} (d_j - \delta_{j,K} \leq Q_{q_{j,K}}), & \text{otherwise},
\end{cases} \tag{2.1}
\]

with \( \delta_{j,k} = \sum_{k'=1}^{k-1} Q_{q_{j,k'}} \) at each step. For example, suppose an event is \( Pl \) with \( d_j = 550 \) sec. For the quartiles in ‘NI’, this would be mapped to \( Pl3 \ Pl3 \ Pl2 \).

Now, let

\[ S = \{Pl1, Pl2, Pl3, Pa1, \ldots, Pa4, Sb1, \ldots, Sb4, Sf1, \ldots, Sf4, Rf, Rs, Rd\} \]

be the set of 18 events with quantized lengths. For each UV Pair, we encode the clickstream log \( E_1, \ldots, E_n \) as \( S = (s_1, s_2, \ldots, s_n) \), where each \( s_j \in S \) is chosen according to the specifications given in this section. As we will see next, using this alphabet that incorporates event types and lengths allows us to obtain insights that cannot be gleaned with events alone.

9The other events do not have this issue since they are not related to processing new, incoming information.
Moving forward, for comparison purposes, we will refer to an event with length 1 as “short,” 2 as “medium,” 3 as “medium-long,” and 4 as “long.”

2.3.3 Motif Extraction

Using the event-type specification, we were able to identify short, recurring sub-sequences within user video-watching behavior, *i.e.*, behavioral *motifs*. As we will see, these motifs capture fundamental characteristics such as reflecting on, skipping over, or revising material. We will also see that some of these motifs are significantly associated with student CFA scores, allowing them to be used for individualization in Chapter 4.

We make use of the MEME Suite software package [11] for motif extraction. MEME has been applied in bioinformatics for motif identification in sequences of nucleotides and amino acids. We turn meme MEME to be applicable in our setting.

Model and Algorithm

The underlying algorithm for motif extraction is based on a probabilistic mixture model, where the key assumption is that each subsequence is generated by one of two components: a position-dependent motif model, or a position-independent background model. Under the motif model, each position $j$ in a motif is described by a multinomial distribution, which specifies the probability of each character (i.e., each $s \in S$ from Section 2.3.2) occurring at $j$. The background model is a multinomial distribution specifying the probability of each character occurring, independent of the positions; we employ the standard background of a 0-order Markov Chain. A latent variable is assumed that specifies the probability of a motif occurrence starting at each position in a given sequence [11].

Motif extraction is formulated as a maximum likelihood estimation over this model, and an expectation-maximization (EM) based algorithm is used to maximize
the expectation of the (joint) likelihood of the mixture model given both the data (i.e., the sequences) and the latent variables. We use the standard dirichlet prior based on character frequencies for EM.

**Extraction**

Each UV Pair’s clickstream sequence is encoded using the 24-character protein alphabet [11]. To do this, we choose the first 18 non-ambiguous characters \( \mathcal{F} \), and then specify a 1:1 mapping \( S \leftrightarrow \mathcal{F} \). Whereas other work has focused on a single motif width (e.g., at 4 in [94]), we extract those of widths \( w \in \{4, ..., 10\} \) from our datasets, with \( E \)-values (defined below) at most 0.05. We will see that both long and short motifs can be insightful (see Figure 2.8).

For each motif, we obtain its \( E \)-value, its position specific probability matrix (PSPM), and its support:

(i) **\( E \)-value**: The \( E \)-value judges overall significance. It is defined as the fraction of motifs (with the same width and occurrences) that would have higher log likelihood ratio if the sequences had been generated according to the background model.

(ii) **PSPM**: This gives the fraction of times that each character appears in each position of the motif, taken over all sightings of the motif in the dataset. In the following, denote the PSPM for a motif by \( \mathbf{P} = [p_{i,j}] \), where \( p_{i,j} \) is the fraction of times event \( j \) occurs at position \( i \).

(iii) **Support**: For each motif, we obtain the fraction of sequences (FS) in which it occurs, i.e., its support across sequences, as well as the number of videos it appears in. We also obtain \( \text{FS0} \) and \( \text{FS1} \) as the fraction of non-CFA and CFA sequences in which the motif appears, respectively.
**Representation**

At each position $i$, we consider all events $j$ with $p_{i,j} \geq 0.25$. Formally, let $\mathcal{A}_i$ be the sequence of indices into the event set $\mathcal{S}$ for $i$, arranged such that $p_{i,\mathcal{A}_i(k)} \geq p_{i,\mathcal{A}_i(k+1)}$ and $p_{i,\mathcal{A}_i(k+1)} \geq 0.25 \ \forall k$. Then, there are three cases on the way $i$ is represented:

- If $|\mathcal{A}_i| > 1$, $i$ is represented as $[\mathcal{S}_{\mathcal{A}_i(1)} \mathcal{S}_{\mathcal{A}_i(2)} \cdots]$.
- If $|\mathcal{A}_i| = 1$, then the square brackets are omitted, with just $\mathcal{S}_{\mathcal{A}_i}$ displayed.
- If $\mathcal{A}_i = \emptyset$, then $i$ is displayed as ‘⋆’ to indicate that this position was taken by a variety of events, none of which occurred even 25% of the time.

For example, the sequence $[\text{P12 P13}] \text{ Pa1 } \star [\text{Sf1 Sf2 Sf4}]$ is of length 4, with the first position being either P12 or P13 at least 50% of the time (P12 at least as often as P13), the second position being Pa1 at least 25% of the time, the third position being any event, and the last being either Sf, Sf2, or Sf4 at least 75% of the time.

**Mixed-Effects Modeling**

To relate video-watching sequences to quiz performance, we seek to quantify the effect that each motif has on whether a UV Pair will have a CFA or a non-CFA response. In fitting such a model, it is important to account for the fact that the individual students and videos can affect the CFA result, since each appears multiple times in the dataset. Hence, for each course, we fit a logistic mixed-effects model [12] to the CFA score, with the frequencies of the motifs treated as fixed effects and the specific user IDs and video IDs as random effects. We denote the fixed-effects matrix is $\mathbf{A} = [a_{s,m}]$, with $a_{s,m}$ as the number of times motif $m$ appears in sequence $s$.

If the $p$-value for a motif in this model is low, we can conclude that the motif has a significant effect on CFA score independent of specific videos and students. To

---

[10]: With 18 different events, a threshold of 25% is roughly 5 times the expected occurrence from a uniform random selection of events.
Figure 2.5: ECDFs of the number of sequences that each motif appears in, for both CFA and non-CFA. The supports are consistent between the CFA and non-CFA groups in each course.

obtain a measure of effect size for each motif, we convert their coefficients from the fitted model (which are expected changes in the log-odds of CFA) to the expected change in CFA probability ($\Delta c$) for each additional occurrence of the motif.

2.3.4 Results

We obtained 87 and 123 motifs from ‘FMB’ and ‘NI’, respectively, which are the subject of the following analysis.

Motif Supports

We first analyze how the motif supports vary across sequences, videos, and students. We find that the supports are reasonably high across videos and students, but that each individual UV Pair tends to not exhibit many motifs.

Sequences. In Figure 2.5, the Empirical CDF (ECDF) of the fraction of sequences that each motif appears in is plotted, for both CFA and non-CFA, considering all sequences with at least one motif. In each course, the supports are similar: for ‘FMB’, each motif appears in 7.1% of the non-CFA sequences on average, and 7.7% of the CFA; for ‘NI’, each appears in 5.9% for both CFA and non-CFA. Considering
The fixed-effects matrices $A$, then, less than 8% of their entries are non-zero. In both courses, the motifs with largest support (first row in Figure 2.8(a) and (b)) appear in $>25\%$ of the sequences.

**Students.** Figure 2.6 gives the ECDF of the fraction of students that trigger each motif at least once (i.e., across all videos the student watched), over students that trigger at least one motif. We see that more users exhibit more motifs in CFA than in non-CFA, for both courses: for ‘FMB’ (resp. ‘NI’), each motif is on average triggered by 15.2% (resp. 12.8%) of users in the CFA sequences, and only 11.1% (resp. 9.9%) in non-CFA.

**Videos.** In Figure 2.7, we show the ECDF of the number of videos that each motif occurred in at least once and at least 10 times (i.e., across all students who saw the video). Overall, CFA has higher support than non-CFA over videos. We also see that the supports decrease for higher thresholds, e.g., for ‘FMB’, while the top 20% of motifs appear in at least 67 videos for CFA, this drops to only 18 videos considering at least 10 occurrences.

We will turn now to analyze the specific motifs, and identify patterns associated with quiz performance. Given that the matrix $A$ is sparse (even among those se-
Figure 2.7: ECDFs of the number of videos that each motif appears in, across both CFA and non-CFA sequences. In both courses, CFA sequences have a higher support for motifs across videos.

sequences that have at least one motif), note that we will move to a more appropriate model for online CFA prediction for individual videos in Section 2.5.

Individual Motifs

We inspect patterns in the most significant of the 210 extracted motifs. This list is obtained by applying the following procedure. First, noticing that all motifs contain play (Pl) events, we group them into categories based on the most recurring alternate event, leading to four groups. Then, within each category, we consider each motif that either (i) has one of the top-5 highest supports or (ii) has a significant p-value (≤ 0.1) returned from the mixed-effects model. Finally, if one motif is a subsequence of another, then we remove the one that has lower support or is less significant.

Figure 2.8 gives the representative sample of these motifs that are mentioned in the following discussion, by group. Each motif is assigned an ID consisting of its group and number (e.g., Pa II in ‘FMB’ is motif P12 Pa4 P12 Pa4). Figure 2.9 visualizes the key properties exhibited by each group.

Overview. The motifs exhibit many similar structural attributes, which occur in spite of the fact that the encoding quantiles are different for each event and course.
(see Figure 2.4). Also, since MEME finds ungapped motifs (i.e., those existing as exact matches in the data, without a separate layer of similarity matching), these identified behaviors exist exactly in the sequences, contrary to other work [94] which has resorted to approximate string searching. The motifs in the Pa (pause) group have the largest supports (FS) overall (≥ 10% mostly), which is consistent with the fact that there are less skip and ratechange events in the datasets (see Figure 2.4(b)).

In what follows, I will present our most interesting observations for each group.

**Reflecting (Pa)**

The occurrence of play together with pause indicates that lectures are generally thought-provoking, causing students to reflect on material they just saw (see Figure 2.9(a)). In both courses, the events forming the motifs in this group cover the entire range from short to medium-long plays (Pl1 – Pl3) interspersed with short to long pauses (Pa1 – Pa4).

The motifs with highest supports in ‘FMB’ and ‘NI’ – Pa I – can be viewed as sequences of medium to medium-long plays with medium-long to long pauses in-between. This behavior is not significantly associated with CFA or non-CFA in either case, though (p-value > 0.1). Motif Pa III in ‘FMB’ is different than these in that it has a short play interspersed too, and it is significantly correlated with an increase in the chance of CFA (p-value < 0.02, Δc = +4.89%). This may indicate that a student pausing longer relative to the plays in-between is an effective strategy in ‘FMB’. Motifs Pa II in ‘FMB’ and ‘NI’, with medium plays followed by long pauses, also do not differentiate between the groups (p-value > 0.1).

The comparison between Pa IV in ‘FMB’ and Pa III in ‘NI’ is particularly interesting. Both of these motifs are short pauses and plays interspersed, indicating a tendency to reflect frequently on a small amount of material at a time. While in ‘FMB’ it is significantly associated with an improvement in CFA (p-value < 0.01,
<table>
<thead>
<tr>
<th>Group</th>
<th>Motif</th>
<th>E-value</th>
<th>FS (%)</th>
<th>FS0 (%)</th>
<th>FS1 (%)</th>
<th>∆c (%)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pa</td>
<td>I [P2 P3] [Pa4 Pa3] [P12 P1] [Pa2 Pa3] P12 Pa3 P12 [Pa2 Pa3] P13</td>
<td>5E-64</td>
<td>28.5</td>
<td>26.2</td>
<td>29.5</td>
<td>+0.68</td>
<td>0.624</td>
</tr>
<tr>
<td></td>
<td>II P12 Pa4 P12 Pa4</td>
<td>2E-06</td>
<td>13.2</td>
<td>13.2</td>
<td>13.3</td>
<td>-1.81</td>
<td>0.145</td>
</tr>
<tr>
<td></td>
<td>IV P11 Pa1 P11 Pa1 P11</td>
<td>7E-13</td>
<td>7.75</td>
<td>6.31</td>
<td>8.38</td>
<td>+5.80</td>
<td>6.2E-3**</td>
</tr>
<tr>
<td>Sb</td>
<td>I Sb3 [P12 P1] [Sb2 Sb3] P12 Sb2 P12 [Sb2 Sb3] [P13 P12]</td>
<td>0</td>
<td>10.2</td>
<td>8.86</td>
<td>10.8</td>
<td>+0.32</td>
<td>0.872</td>
</tr>
<tr>
<td></td>
<td>II Sb4 P11 Sb4 P12 [Sb3 Sb4]</td>
<td>8E-05</td>
<td>4.64</td>
<td>3.75</td>
<td>5.03</td>
<td>+7.52</td>
<td>6.8E-3**</td>
</tr>
<tr>
<td></td>
<td>III Sb3 P12 Sb3 [P13 P12]</td>
<td>0</td>
<td>3.54</td>
<td>3.08</td>
<td>3.74</td>
<td>+4.67</td>
<td>0.0638</td>
</tr>
<tr>
<td>Sf</td>
<td>I [P12 P1] [Sf3 Sf2] [P11 P12] Sf2 [P11 P12] Sf1 [P12 P11] [Sf2 Sf1]</td>
<td>0</td>
<td>9.50</td>
<td>9.56</td>
<td>9.47</td>
<td>+1.29</td>
<td>0.591</td>
</tr>
<tr>
<td></td>
<td>II [Sf4 Sf4] [P11 P12] [Sf3 Sf4] [P11 P12] [Sf3 Sf4]</td>
<td>0</td>
<td>7.59</td>
<td>7.61</td>
<td>7.58</td>
<td>-3.36</td>
<td>0.130</td>
</tr>
<tr>
<td>Rf</td>
<td>I P3 [Rf R] [P12 P1] Rf Pl1 P12 [P12 P1] Rf Pl1 P12</td>
<td>0</td>
<td>4.55</td>
<td>3.89</td>
<td>4.84</td>
<td>+1.62</td>
<td>0.539</td>
</tr>
<tr>
<td></td>
<td>II Rf R [Pl1 Pl2] Rf P13</td>
<td>1E-70</td>
<td>1.77</td>
<td>1.22</td>
<td>2.00</td>
<td>+9.30</td>
<td>0.039*</td>
</tr>
</tbody>
</table>

(a) Motifs for ‘FMB’.

<table>
<thead>
<tr>
<th>Group</th>
<th>Motif</th>
<th>E-value</th>
<th>FS (%)</th>
<th>FS0 (%)</th>
<th>FS1 (%)</th>
<th>∆c (%)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>II P12 Pa4 P12 Pa4</td>
<td>2E-44</td>
<td>14.3</td>
<td>15.9</td>
<td>13.7</td>
<td>-2.71</td>
<td>0.105</td>
</tr>
<tr>
<td></td>
<td>III P11 Pa1 P11 Pa1 P11 Pa1 P11 [P11 P13]</td>
<td>0</td>
<td>11.7</td>
<td>12.7</td>
<td>11.4</td>
<td>-5.29</td>
<td>0.036*</td>
</tr>
<tr>
<td>Sb</td>
<td>I Sb3 Sb4 [P12 P13] Sb2 Sb3 Sb2 [P13 P12]</td>
<td>0</td>
<td>9.17</td>
<td>9.47</td>
<td>9.07</td>
<td>+0.74</td>
<td>0.787</td>
</tr>
<tr>
<td></td>
<td>II Sb2 P12 Sb3 P12 [Sb2 Pa3]</td>
<td>6E-63</td>
<td>6.05</td>
<td>5.76</td>
<td>6.15</td>
<td>-5.93</td>
<td>0.065</td>
</tr>
<tr>
<td></td>
<td>III Sb2 [P12 P1] [Sf2 Sf3] Pl1 [Sf3 Sf2]</td>
<td>0</td>
<td>7.47</td>
<td>4.76</td>
<td>6.07</td>
<td>+9.63</td>
<td>7.8E-3**</td>
</tr>
<tr>
<td>Sf</td>
<td>I [P3 P1] [Sf3 Sf2] [P11 P12] Sf2 [Sf3 Sf2]</td>
<td>0</td>
<td>7.17</td>
<td>7.47</td>
<td>7.04</td>
<td>-5.97</td>
<td>0.024*</td>
</tr>
<tr>
<td></td>
<td>II [Sf3 Sf4] [P11 P12] [Sf3 Sf4] [P11 P12] [Sf3 Sf4]</td>
<td>0</td>
<td>7.76</td>
<td>8.95</td>
<td>7.36</td>
<td>+6.45</td>
<td>0.085</td>
</tr>
<tr>
<td>Rf</td>
<td>I P11 Rf Rf P12 Rf P12 Rf</td>
<td>0</td>
<td>9.21</td>
<td>8.58</td>
<td>9.43</td>
<td>+8.07</td>
<td>9.8E-3**</td>
</tr>
<tr>
<td></td>
<td>II Rf R [Pl1 P12] Rf P11 Rf Rf Pl1 P12 Rf Pl3</td>
<td>0</td>
<td>3.86</td>
<td>4.61</td>
<td>3.61</td>
<td>-10.8</td>
<td>0.016*</td>
</tr>
<tr>
<td></td>
<td>III Rf R [Pl1 P12] Rf Pl1 P12 Rf P13</td>
<td>7E-16</td>
<td>2.53</td>
<td>3.14</td>
<td>2.33</td>
<td>-8.11</td>
<td>0.089</td>
</tr>
</tbody>
</table>

(b) Motifs for ‘NI’.

Figure 2.8: Representative sample of motifs identified for each course. Each motif is grouped by the dominant event it contains outside of P1. FS is the fraction of sequences over both CFA and non-CFA, while FS0 and FS1 are for the separate cases. ∆c is the estimated change in the probability of success (CFA) for every additional occurrence of the motif, and the p-value (p) is the significance of ∆c (a . indicates p ≤ 0.1, a * indicates p ≤ 0.05, and a ** indicates p ≤ 0.01).
Figure 2.9: Illustration of the key video-watching behaviors exhibited by each of the four motif groups in Figure 2.8. For brevity, we omit variations that exist between the individual motifs within the groups. In each case, the horizontal axis represents the video length, as in Figure 2.3. For (b) and (c), horizontal jumps represent skips with lengths relative to the distance, whereas the vertical jumps in (b) just indicate continuity.

\[ \Delta c = +5.80\% \], in ‘NI’, it is associated with a decrease in the chance of CFA (\( p \)-value < 0.05, \( \Delta c = -5.29\% \)). Short pauses in ‘NI’ may be a sign of unresolved confusion.

Revising (Sb)

From the six motifs in the Sb group, we identify two interesting, recurring subsequences: P12 Sb3 P12 Sb3 (Sb I and III in ‘FMB’, and Sb I in ‘NI’), and P12 Sb2 P12 Sb2 (Sb III in ‘NI’). Roughly speaking, each of these is associated with \textit{playing for a length of video, and then revising some or all of that content} (see Figure 2.9(b)).

To see this, consider the ranges of P1 and Sb from Figure 2.4 associated with these subsequences: for ‘FMB’, P12 covers 14 to 68 sec, and Sb2 to Sb3 covers 18 to 73 sec; for ‘NI’, these ranges are 12 to 71 sec and 13 to 55 sec. The play and skip ranges are closely overlapping in each case. Taking the extreme ends of each range, they are
associated with skipping back anywhere from 1 min below the starting play point to 50 sec after it, which are local considering the video lengths.

Note that 2 of the 4 motifs containing these behaviors are significantly correlated with an increase in CFA probability ($p$-value < 0.07, $\Delta c > +4.0\%$). The fact that Sb II of ‘FMB’ has the highest $\Delta c = +7.52$ may also indicate that *revising more than what was just played* can further increase the chance of success, given the presence of long skip backs (Sb4) here.

We also considered the number of skip backs originating at each video position across all UV Pairs. We find that the largest origination point of these events is at the end of videos. In particular, out of all Sb events, those originating within 10 sec of the videos’ end constitute roughly 15% of the total in ‘FMB’ for both non-CFA and CFA sequences (as a reference, if we take the highest location of Sb for each video outside of the last 10 sec, these positions contain roughly 4% of the total for non-CFA and CFA). This, combined with the motifs suggesting improvement when revisions occur, implies that those students who are revising *multiple times* before answering a quiz have more success.

The notable exception to this is Sb II in ‘NI’. Here, revision is associated with a *decrease* in the chance of CFA ($\Delta c = -5.93\%$). Similar to Pa III discussed before, this may be an indicator of excessive confusion in this course.

**Skimming (Sf)**

In both of the courses, the motifs in the Sf group are primarily medium to long skips forward with short to medium plays in-between. Further, the skips are longer than the plays occurring before and after; comparing the lengths of Pl and Sf events in Figure 2.4, we see that for both courses, range $Q_j$ to $Q_{j+1}$ for Sf is always larger than $Q_{j-1}$ to $Q_j$ for Pl. This recurring behavior can then be interpreted as *skimming*

\[\text{We assume a default playback rate as an approximation.}\]
through the material quickly with less exposure to the material (see Figure 2.9(c)).

We find that 3 of these 6 motifs are significant in favor of non-CFA ($p$-value < 0.07, $\Delta c < -4.0\%$), in contrast to the results from Section 2.2 where the total number of skip forwards exhibited by a UV Pair was not found to be correlated with the CFA probability. This difference underscores an advantage of considering the clickstream sequences, rather than computing aggregate quantities to summarize them.

Also, $S_b$ and $S_f$ occurring together in a motif (e.g., $S_f$ II in ‘FMB’) can possibly be interpreted as skipping forward with caution. Still, we find that this is close to being significant in favor of non-CFA ($p$-value = 0.13). There is an exception to the generally negative correlations in the $S_f$ group, though: $S_f$ II in ‘NI’, where skimming is associated with an increase in CFA probability ($p$-value < 0.1, $\Delta c = +6.45$). With short plays in-between long skips forward, this is fast skimming, and can be explained by the fact that in some cases, a student will already be familiar with the more basic ‘NI’ material.

Speeding ($R_f$)

$R_f$ I in ‘NI’ indicates that viewing the material at a faster than default rate, i.e., speeding, is significantly associated with an increase in CFA probability ($p$-value < 0.01, $\Delta c > +8.0\%$). The other motifs making up the $R_f$ groups for each course have students returning to the default rate ($R_d$), indicating they are slowing down for important content (see Figure 2.9(d)). In ‘FMB’, this is positively associated with CFA in one case ($R_f$ II, with $\Delta c = +9.3\%$), whereas in ‘NI’, it is significantly associated with a decrease in CFA score in both cases ($R_f$ II and $R_f$ III, with $\Delta c < -8.0\%$). In ‘NI’, slowing the rate back down to the default could be a sign that a student saw something confusing, but did not not take the time to e.g., reflect or revise to clear up the confusion.
2.3.5 Key Messages

Overall, we draw a few conclusions from the motifs.

Motif Groups

There are four main groups:

- **Reflecting, i.e.,** pausing to reflect on the video material repeatedly. If the time spent reflecting is not *too* long relative to the time spent watching, this tends to be correlated with a higher chance of success on the quiz. At the same time, if the pausing is very short, it can indicate unresolved confusion.

- **Revising, i.e.,** repeated revision of the video content just watched. This tends to be correlated with an increase in the chance of success.

- **Skimming, i.e.,** skipping through video material quickly. This tends to be associated with a lower chance of success, even when done with caution.

- **Speeding, i.e.,** watching the video at a faster than default rate and slowing down at certain times. Different variations are associated with different impacts on the chance of success.

Importance of CFA Associations

Some motifs are significantly correlated with substantial changes in the probability of CFA, independent of the specific videos and/or students (from the change in CFA probability $\Delta c$ in Figure 2.8, the increases can be as high as 9%, and the decreases as low as 10%). For each motif, the direction of the association is particularly important, because in many cases either would be intuitive. For example, a revising motif could presumably come from a student reinforcing material in the video prior to taking the quiz (in line with an increase in CFA probability) or from excess confusion caused
by the material in the video (in line with a decrease in CFA probability), but the results indicate the former tends to be more likely in these courses. As another example, skimming could come from a student believing confidently that he/she is already familiar with the content in a video, which could intuitively be either a correct (increase in CFA probability) or an incorrect (decrease in CFA probability) perception, but results favor the latter.

**Importance of Lengths and Durations**

We emphasize the importance of having included the lengths/durations, in addition to the events, in our framework in order to make these conclusions. For instance, the sequence Pl Sb Pl Sb identified in [94] cannot be associated with revising, because it is not clear how far back the student has skipped relative to having played in-between. In the same way, Pl Sf Pl Sf cannot be concluded as skimming, because the lengths of play and skip are not indicated in the model. Also, even small changes in the motif lengths can affect significance; for example, in ‘FMB’, while Pa I is associated with CFA, Pa II is not, and Pa II is almost a subsequence of Pa I.

### 2.4 CFA Prediction via Aggregate Quantities

At this point, the first two research questions posed at the beginning of Section 2.1 – related to representation and correlation of video-watching behaviors – have been investigated. Now, we move to the third question, which seeks to use the findings presented thus far to enhance performance prediction for MOOC. We begin in this section by proposing and evaluating CFA prediction algorithms based on the aggregate quantities defined in Section 2.2.

**Definitions.** In general, let \( n \in \Omega \) denote entry-instance \( n \) in the set of all entries \( \Omega \) that form the full dataset. We index users (students) by \( i \) and quizzes (videos)
by $j$; each User-Video Pair is associated with a particular user $u(n)$, quiz $q(n)$, CFA score $c_n \in \{0, 1\}$ (1 is CFA, 0 is non-CFA), and algorithm prediction $\hat{c}_n \in [0, 1]$. We also write $n = e(i, j)$ to denote the entry $n$ associated with user $i$ and quiz $j$, where $e : (i, j) \rightarrow \Omega$. For evaluation, we generate training and test sets as subsets of $\Omega$ through a procedure described in Section 2.4.4; the training set $\Omega_T$ and test set $\Omega_E$ are always chosen such that $\Omega_T \cap \Omega_E = \emptyset$.

2.4.1 Our Algorithms Using Clickstream Data

Motivated by the correlation findings in Section 2.2, our prediction algorithms determine suitable intervals/sets of values (referred to generally as intervals) for each of the quantities by analyzing the densities over $\Omega_T$, estimate the CFA probabilities within each interval, and use them as learning features. In Section 2.4.4, this will be seen to improve performance relative to standard prediction algorithms for all metrics and dataset partitions tested.

Interval Extraction

Let $\Omega_{T}^c$ be the subset of $\Omega_T$ belonging to class $c \in \mathcal{O} = \{0, 1\}$, i.e., $\Omega_{T}^c = \{n \in \Omega_T : c_n = c\}$. Also, let $f \in \mathcal{V}$ denote aggregate quantity $f$ in the set of quantities $\mathcal{V} = \{1, ..., 8\}$. In determining suitable intervals over $\Omega_T$, we group the quantities into three types:

(i) Continuous (1 – 3, 5): For each continuous $f$, we approximate the probability density function of each class $c$ over $\Omega_T^c$ with a Kernel Density Estimator (KDE):

$$ p^c_f(v) = \frac{1}{|\Omega_T^c|\eta} \sum_{n \in \Omega_T^c} \kappa \left( \frac{v - v_n^f}{\eta} \right), \quad (2.2) $$

12we do not use 9 because it did not have significant intervals.
where \( v^f_n \) is the value that quantity \( f \) takes for entry \( n \), \( \eta \) is the bandwidth of the estimator, and \( \kappa(\cdot) \) is the kernel function \[79\]. Here, we use the standard Gaussian Kernel, and fit the estimator for values \( v \in [0, u_f] \) (the upper bound controls for outliers). Then, we find the intersection points between \( p^0_f(v) \) and \( p^1_f(v) \) as the boundaries between the intervals for \( f \). More formally, define the ordered set

\[
\mathcal{I}_f = \{0\} \cup \{v : p^0_f(v) = p^1_f(v)\} \cup \{u_f\} 
\]  

Then, there are \(|\mathcal{I}_f| - 1\) intervals, where interval \( h \in \{1, ..., |\mathcal{I}_f| - 1\} \) spans the range \( \mathcal{B}_h^f = [\mathcal{I}_f(h), \mathcal{I}_f(h + 1)] \).

(ii) Discrete (4 & 8): For discrete \( f \), we compute the empirical probability mass function of each group \( c \), \( p^f_c(v) \), over \( \Omega_T^c \) for values \( v \in \{0, ..., u_f\} \). Then, we find the values \( v \) at which a change occurs in the class that has more density between \( v \) and \( v + 1 \). More formally, we let

\[
\mathcal{I}_f = \{0\} \cup \{v : p^0_f(v) \leq p^1_f(v) \land p^0_f(v + 1) \geq p^1_f(v + 1)\} \cup \{u_f\} 
\]  

The \( \geq \) and \( \leq \) symbols used in this way imply \( > \) and \( < \), or \( < \) and \( > \). The interval boundaries are defined by these changes, \( i.e., \mathcal{B}_h^f = \{\mathcal{I}_f(h), ..., \mathcal{I}_f(h + 1) - 1\} \).

(iii) Binary (6 & 7): Though these two features take on continuous values, we saw in Section 2.2 that it is more informative to group each of them into two sets: \( \mathcal{B}_0^f = \{G_f\} \) and \( \mathcal{B}_1^f = \mathbb{R}_{\geq 0} \setminus \mathcal{B}_0^f \), where \( G_6 = 1 \) and \( G_7 = 0 \).

Success Estimates

We now compute the CFA estimates for each \( \mathcal{B}_h^f \). First, the total occurrences of group \( c \) in \( h \) over \( \Omega_T \) is

\[
O^c_f[h] = \sum_{n \in \Omega_T} \mathbb{I}_{\{c_n = c, v^f_n \in \mathcal{B}_h^f\}},
\]  

46
where $\mathbb{I}$ is the indicator function. The corresponding fraction is $d_f^c[h] = O_f^c[h] / |\Omega_T|$. Then, letting $O_f[h] = O_f^0[h] + O_f^1[h]$, we apply Laplace’s rule of succession \cite{79} to compute the estimated probability of a new element $n$ (i.e., those in $\Omega_E$) with $v_n^f \in B_h$ having $c_n = c$:

$$\hat{p}_f[h] = \frac{r_f[h] \cdot O_f[h] + 1}{O_f[h] + 2},$$

(2.6)

where $r_f[h] = d_f^1[h] / (d_f^0[h] + d_f^1[h])$ is the fraction of density in $h$ that is of the positive class.\footnote{The terms of 1 and 2 in the numerator and denominator of $\hat{p}_f[h]$ are required under Bayesian theory to generate the correct estimate over a normal prior.} For the examples with $v_n^f \notin [0, u_f]$, we set $\hat{p}_f[h] = 0.5$ (i.e., the population average).

Finally, to account for the fact that there can be high variation in the number of samples for each interval, we apply a shrinkage estimator as follows:

$$\bar{p}_f[h] = \frac{\hat{p}_f[h] \cdot O_f[h] + 0.5\sigma|\Omega_T|}{O_f[h] + |\Omega_T|},$$

(2.7)

where $\sigma$ is a parameter controlling the weight of the population average. In this way, the success estimates are adjusted based on the sample sizes of the corresponding interval to reduce the possibility of overfitting based on a small number of samples. This is particularly useful for the continuous features, where $\eta$ trades off the bias and variance of the KDE \cite{79}; as we will see in Section 2.4.4 including $\sigma$ allows the estimators to risk choosing a lower $\eta$ (higher variance), thereby generating a larger number of smaller-width intervals, since the $\hat{p}_f[h]$ are adjusted accordingly. To evaluate this tradeoff, we will consider two separate instances: VID-A, which uses $\bar{p}_f[h]$ for the success estimates, and VID-N, which uses $\hat{p}_f[h]$ instead.

**SVM Classification**

These $\bar{p}_f[h]$ (or $\hat{p}_f[h]$) are used as features in a Support Vector Machine (SVM) classifier. We choose SVM because it can readily generate complex decision boundaries
through application of different kernel functions \[61\]. We visualize the design matrix \( X \) for the SVM scheme in Figure 2.10: with \( I \) users, \( J \) quizzes, and \(|V| = 8\) clickstream quantities, each instance \( n \) is described by an \( I + J + |V| \) dimensional feature vector \( x_n \), with indicator features for user and quiz, and the CFA probability estimates for each \( f \). For the estimates, we find interval \( h_f \) such that \( v^f_n \in B^f_h \) and then use \( \hat{p}_f[h_f(n)] \) (or \( \hat{p}_f[h_f(n)] \)) as the corresponding feature value. Then, the following optimization problem is solved, adjusted from [79]:

\[
\begin{align*}
\minimize_{w, w_0, \epsilon} & \quad \frac{1}{2}||w||^2 + C \sum_{n \in \Omega_T} \epsilon_n \\
\text{subject to} & \quad c^n_n (\kappa(w, x_n) + w_0) \geq 1 - \epsilon_n, \ \forall n \in \Omega_T \\
& \quad \epsilon_n \geq 0, \ \forall n \in \Omega_T
\end{align*}
\]

(2.8)

where \( c^n_n = 2(c_n - 0.5) \) (i.e., \(-1 \) or \(+1\) instead of \(0\) or \(1\)). Here, we use a polynomial kernel of the form \( \kappa(z, z') = (\pi \cdot z^T z' + a_0)^d \), because (i) it readily generates product features of degree \( d \) to help capture interaction terms between the features and (ii) it was seen to give better performance than other kernel choices (such as Gaussian).

With the resulting \( w, w_0, \) and \( \epsilon \), the individual probabilities \( \hat{c}_n \in [0, 1] \) are generated using the standard Platt scaling procedure implemented in [83].
For VID-A, \(d, C, \pi, a_0, \sigma, \) and \(\eta\) are parameters that must be tuned through cross validation (described in Section 2.4.4). For VID-N, all except \(\sigma\) must be tuned. We set the \(u_f\) based on our intuition for each aggregate quantity: \(u_1 = u_3 = 2, u_2 = u_5 = 1, u_4 = 10,\) and \(u_8 = 5.\)

2.4.2 Standard Algorithms in Big Data

For comparison, we will employ a number of standard algorithms that have been applied for prediction in traditional education settings and leverage only performance data. We describe those now.

Naive

This predicts the global average over \(\Omega_T\) for all \(n,\) i.e., \(\hat{c}_n = \sum_{n' \in \Omega_T} c_{n'} / |\Omega_T|\). This most naive predictor will only serve as a benchmark for measuring incremental improvement for all other algorithms.

Biases

This includes a bias for each user and quiz, and is equivalent to the Rasch model in Item Response Theory [14] with a global bias term. Letting \(b_{u(n)}\) and \(b_{q(n)}\) be the biases for \(u(n)\) and \(q(n), b\) be the vector of biases, and \(\mu\) be a global term, we solve

\[
\arg\min_{\mu, b} \sum_{n \in \Omega_T} -\ln \Phi(\hat{c}_n' \cdot c_n') + \frac{\lambda_B}{2} ||b||^2,
\]

where \(\hat{c}_n' = 2(\hat{c}_n - 0.5), c_n' = 2(c_n - 0.5),\)

\[
\hat{c}_n = \mu + b_{u(n)} + b_{q(n)},
\]

\(\lambda_B\) is the regularization parameter, and \(\Phi(\cdot)\) is the sigmoid function \(\Phi(z) = 1/(1 + e^{-z}).\) Here, \(\lambda_B\) is a parameter to be tuned.
Matrix Factorization (MF)

This model includes a latent factor vector of dimension $K_M$ for each user, $\mathbf{u}_i \in \mathbb{R}^{K_M}$, and quiz, $\mathbf{q}_j \in \mathbb{R}^{K_M}$, in addition to the biases from the previous model. It is a common collaborative filtering method that modifies conventional Singular Value Decomposition (SVD) to learn only over known training instances [70, 13, 72]. Letting $\mathbf{U} = [\mathbf{u}_i]$ and $\mathbf{Q} = [\mathbf{q}_j]$ be the matrix of user and quiz factors, respectively, we solve

$$\arg\min_{\mu, \mathbf{b}, \mathbf{U}, \mathbf{Q}} \sum_{n \in \Omega_T} - \ln \Phi (\hat{c}'_n \cdot c'_n) + \frac{\lambda_B ||\mathbf{b}||^2 + \lambda_M (||\mathbf{U}||^2 + ||\mathbf{Q}||^2)}{2},$$

(2.11)

where $\hat{c}'_n$ and $c'_n$ are defined as in Biases, and

$$\hat{c}_n = \mu + b_{u(n)} + b_{q(n)} + \mathbf{u}^T_{u(n)} \mathbf{q}_{q(n)}.$$  

(2.12)

$\lambda_B$, $\lambda_M$, and $K_M$ must be tuned.

K Nearest Neighbor (KNN)

For user $i$, this uses the CFA scores of the set of $K_N$ users $\mathbf{U}_i$ who have the most similar quiz results to $i$ for prediction. We determine $\mathbf{U}_i$ for all $i$ as in [101]: (i) compute the Pearson correlation coefficient $\rho_{i,i'}$ between $i$ and all other users $i'$ using the performances on the set of quizzes $\mathcal{J}_{i,i'}$ common to the pair in $\Omega_T$, (ii) shrink the correlations according to $\bar{\rho}_{i,i'} = \frac{|\mathcal{J}_{i,i'}| \rho_{i,i'}}{|\mathcal{J}_{i,i'}| + \alpha}$, (iii) apply the sigmoid mapping $\tilde{\rho}_{i,i'} = \Phi (\delta \bar{\rho}_{i,i'} + \gamma)$, and (iv) define $\mathbf{U}_i$ as the set of $K_N$ users $i' \neq i$ with maximum $|\tilde{\rho}_{i,i'}|$. Then,

$$\hat{c}_n = \frac{\sum_{i' \in \mathcal{U}'_{u(n)}} \tilde{\rho}_{u(n),i'} \cdot y_e(i',q(n)) + m_{u(n)} \beta}{\sum_{i' \in \mathcal{U}'_{u(n)}} |\tilde{\rho}_{u(n),i'}| + \beta},$$

(2.13)

where $\mathcal{U}'_{u(n)}$ is the set of only those users $i' \in \mathbf{U}_{u(n)}$ with $e(i', q(n)) \in \Omega_T$, and $m_i$ is the mean score of user $i$ over $\Omega_T$. Here, $\alpha, \delta, \gamma, \beta$, and $K_N$ are parameters to be tuned.
2.4.3 Metrics

We will use three standard classifier metrics to compare the algorithms:

**Accuracy.** Let $\tilde{c}_n \in \{0, 1\}$ denote the rounded output of the prediction $\hat{c}_n$ for entry $n$. The accuracy is the fraction of these in the test set that are correct, i.e.,

$$\sum_{n \in \Omega_E} 1\{\tilde{c}_n = c_n\} / |\Omega_E|.$$

**RMSE.** Unlike accuracy, the Root Mean Squared Error (RMSE) uses $\hat{c}_n$ directly, and is evaluated as

$$\sqrt{\sum_{n \in \Omega_E} (c_n - \hat{c}_n)^2 / |\Omega_E|}.$$

**AUC.** This measures the Area Under the Receiver Operating Characteristic (AU-ROC) curve of the classifier, where the ROC plots the tradeoff between the true and false positive rates [79]. AUC can also be seen as the probability that the classifier will rank a randomly chosen instance in the positive class (i.e., $c_n = 1$) higher than a randomly chosen negative instance.

Note that even one percent improvement in some of these metrics can be substantial. As a reference, for CFA prediction in KDD Cup 2010 there was only 1% improvement in RMSE from the 132nd to the best score on the leaderboard [34]

2.4.4 Evaluation

We will now move to compare the performance of the algorithms on the ‘FMB’ dataset. In doing so, we will find it informative to consider different subsets of the dataset, to see how the performance varies under different conditions. In general, let $\Omega^{u_0,v_0} \subset \Omega$ denote the set consisting of all instances $n$ such that the user $u(n)$ has $\geq u_0$ instances over $\Omega$ (i.e., having answered at least $u_0$ questions) and the video $q(n) \leq v_0$ (i.e., within the first $v_0$ videos of the course).

[14]https://pslcdatashop.web.cmu.edu/KDDCup/LeaderBoard
Implementation Details

**Software.** The Biases and MF algorithms were each implemented with libFM [89] using stochastic gradient descent (SGD) with a small enough step size (0.01) and a large enough number of iterations (8000) such that we observed convergence in all cases. For VID-A and VID-N, the KDE method is implemented through Python’s scikit-learn [83], and the SVM classifier through libSVM [38] with the SMO algorithm. The naive and KNN algorithms were programmed de novo in Python.

**Training and cross validation.** In evaluating the algorithms over $\Omega^{u_0, v_0} = \Omega'$, we use k-fold cross validation (CV) [79] to consider multiple training/test set partitions. We partition $\Omega'$ into $k$ disjoint subsets $\Omega_1, ..., \Omega_k$ such that $\Omega_1 \cup \cdots \cup \Omega_k = \Omega'$. These subsets are formed as follows: letting $\mathcal{N}_{i'} = \{n \in \Omega' : u(n) = i'\}$ (i.e., all instances of user $i'$ in $\Omega'$), we randomly permute $\mathcal{N}_{i'}$ and allocate the $l$th set of $\lfloor |\mathcal{N}_{i'}| / k \rfloor$ instances to $\Omega_l$; the remainder is allocated across the subsets randomly. This process is repeated over all users, and is done to ensure that the entries for each user are spread evenly across the sets, since the $|\mathcal{N}_i|$ tend to be small. With the $k$ subsets in hand, for each $z = 1, ..., k$ each algorithm is trained on $\Omega_T = \Omega' \setminus \Omega_z$ and tested on $\Omega_E = \Omega_z$, and each of the metrics are averaged over the $k$ trials. We set $k = 5$ in our evaluation.

**Parameter tuning.** We handle tuning of continuous and discrete parameters differently in a procedure which we followed closely for each algorithm. The continuous ones were tuned over $\Omega^{20,92}$ using a multi-dimensional grid search procedure [61]. To do this, we first randomly selected 15% of the instances from each subset $\Omega_l$ [15]. Then, the following is performed for each algorithm:

1. Choose initial center points $q_p \in \mathbb{R}$, ranges $r_p \in \mathbb{N}$, and step sizes $s_p \in \mathbb{R}_{>0}$ for each parameter $p$.  

---

[15] In the end, for KNN we used the entirety of $\Omega^{20,92}$ because of its sensitivity to selecting the specific neighbors for each user. For the other algorithms, 15% was not seen to significantly affect the parameter choices.
Algorithm | Parameters
--- | ---
Biases | \( \lambda_B = 0.105 \)
MF | \( K_M = 3, \lambda_B = 0.115, \lambda_M = 0.181 \)
KNN | \( K_N = 30, \alpha = 18.4, \beta = 1.9, \delta = 14.1, \gamma = -5.0 \)
VID-N | \( d = 6, C = 0.011, \pi = 0.608, c_0 = 1.88, \eta = 0.183 \)
VID-A | \( d = 6, C = 0.0062, \pi = 2.38, c_0 = -1.44, \sigma = 0.030, \eta = 0.0186 \)

Table 2.3: Tuned parameters for each of the algorithms.

2. Run 5-fold CV over all combinations in the set \( G = \{2^{q_1-r_1s_1}, \ldots, 2^{q_1+r_1s_1}\} \times \cdots \times \{2^{q_P-r_ps_P}, \ldots, 2^{q_P+r_ps_P}\} \) (for the parameters that can take positive and negative values, \( G = \{q_1-r_1s_1, \ldots, q_1+r_1s_1\} \times \cdots \times \{q_p-r_ps_P, \ldots, q_p+r_ps_P\} \) instead), where \( P \) is the total number of parameters.

3. Set \((q_1, \ldots, q_P) = G_g\), where \( g \) is the index of the combination with the highest accuracy, and set \( s_p \) to \( s_p/\zeta, \zeta > 1 \forall p \).

4. Repeat 1–3 until \((q_1, \ldots, q_P)\) does not change between three successive iterations.

The final \((q_1, \ldots, q_P)\) are taken as the tuned parameters.

Since the behavior of the discrete parameters is not easy to capture in the above procedure, we simply repeat the search over the continuous parameters for each discrete choice, choosing the best overall combination. The final values tested for \( K_M \) and \( d \) were in \( \{1, \ldots, 10\} \), and for \( K_N \) were in \( \{20, \ldots, 40\} \). The performance of matrix factorization in educational settings has been seen to saturate after the first few factor dimensions \( K_m [13] \), and for KNN, the performance did not vary much within \( K_n \) either.

In Table 2.3 we give the tuned parameters for each algorithm that are used to generate the following results. As expected, VID-A has a lower \( \eta \) than VID-N.
Results

We now present an evaluation of the algorithms under different scenarios, followed by additional discussion. In comparing the performance between algorithms, we will consider the percent improvement (PI) of each over the Naive benchmark, which has an accuracy of roughly 0.66 and an RMSE and AUC of 0.5 in each case. To obtain PI, we measure percent increase for accuracy and AUC (higher is better), and percent decrease for RMSE (lower is better). We repeated the CV procedure 5 times for each algorithm in each dataset (i.e., 25 runs each) and averaged the results.

Active users over the full course. We first evaluate the algorithms on \(\Omega^{20,92}\), to only consider the active users who have taken at least 20 questions. Table 2.4(a) tabulates the results for each of the algorithms and metrics, and Figure 2.11(a) shows the PI in each case. Notice that VID-N and VID-A both outperform the three standard algorithms at least slightly for each of the metrics. For accuracy and AUC, the improvement differential is marginal, with an increase of 0.27% (in both cases) from Biases to VID-A and VID-N, respectively, while for RMSE, it is more pronounced, with an increase of 0.89% from Biases to VID-A. Among the standards, the KNN algorithm performs the worst in all cases, which is consistent with results in other work (e.g., [101]). One surprising point is that MF has slightly lower performance than Biases even though it adds factor dimensions, likely due to overfitting. Among the clickstream algorithms, we see that VID-A performs better than VID-N on the accuracy metric but that on RMSE and AUC they are roughly equivalent.

All users over the first two weeks. Next, we evaluate the algorithms on \(\Omega^{0,20}\), to consider all users in the first two course weeks, i.e., the beginning of the course. Table 2.4(b) tabulates the absolute metrics, and Figure 2.11(b) shows the percent improvements. Compared with the previous case, each algorithm has lower performance, which is expected since we have less information to learn from for each user and quiz. Additionally, we see that MF now slightly outperforms Biases, and that VID-A has
Table 2.4: Absolute performance metrics obtained from evaluating the algorithms over different subsets of the course data. Bold denotes the best achieved in each case.

substantially higher performance than VID-N overall; this second point highlights the utility of including both \( \eta \) and \( \sigma \) in the video-watching scheme. Finally, we see that both of the video-watching algorithms again outperform the standard algorithms for each metric. But the remarkable result here is the incremental improvement, which shows an increase of 2.06\%, 2.02\%, and 4.78\% for accuracy, RMSE, and AUC from MF to VID-A. Hence, it is reasonable to conclude that video-watching data is particularly useful for performance prediction early in a course when it is not yet clear which users are active.

All users over the full course. Finally, we perform an evaluation over \( \Omega \) to consider learning on the full dataset. Table 2.4(c) tabulates the absolute metrics, and Figure 2.11(c) shows the percent improvements. Compared with the previous two cases, for both accuracy and AUC the algorithms show lower performance than \( \Omega^{20,92} \) but higher than \( \Omega^{0,20} \); on the other hand, the RMSE improvements are roughly consistent with those from \( \Omega^{0,20} \). The improvement of VID-A compared with the standards is
Figure 2.11: Percent increases in prediction performance relative to the benchmark. VID-N and VID-A outperform the standard algorithms in each case for each metric; however, the incremental gain is the highest when considering VID-A in $\Omega^{0,20}$.

1.55%, 1.91%, and 4.60% for each metric, which is substantial but not as high as for $\Omega^{0,20}$.

2.4.5 Key Messages

The evaluations in this section consider one type of student behavior – video-watching data – to see how it can improve quiz performance prediction. We see that incor-
porating it can achieve gains over standard algorithms, but that the highest benefit comes from applying it early in the course (Figure 2.11(b)) for “quickest detection” of students who may be struggling with particular content. The reason for the high differential in this case is that there is relatively few entries for each individual user, which puts the Biases and Matrix Factorization algorithms at a clear disadvantage since they rely on explicit user models (\(i.e., b_i \) and \(u_i \) from Section 2.4.2). On the other hand, while the clickstream algorithms incorporate user/quiz biases (\(i.e., \) the indicator features in Figure 2.10), they also leverage the video-watching data aggregated over all users to assist in classifying CFA or not.

The sensitivity of the standards to per-user information is further emphasized when considering all users over the entire course (Figure 2.11(c)): the performance improves, because there are more entries for each user, but is not near what is possible when only considering active users, because there are many who only take a few quizzes due to the dropoff rate in MOOC.

As to the specific video-watching algorithms, VID-A was seen to outperform VID-N substantially in the second two dataset partitions considered. Recall that the difference between them is that the smaller \(\eta\) for VID-A causes the KDE for clickstream quantities 1-3,5 to generate more, smaller-width intervals (\(e.g., \) for \(\Omega^{20,92}\) the average number of intervals across these quantities was 13 for VID-A and only 2.4 for VID-N), since \(\sigma\) compensates for overfitting on small sample sizes. Note also that \(\eta\) in VID-A is close to those used in Figure 2.2, except that for the preliminary analysis we focused only on the 2-3 intervals we identified visually.

### 2.5 CFA Prediction via Sequences

Motivated by the early detection capability of the aggregate quantities, we now move to investigate the applicability of sequence representations to CFA prediction.
In Section 2.3, we saw that each individual UV Pair did not exhibit many motifs, which implies they are sparse when it comes to CFA prediction. As a result, rather than using motifs, we will formalize another sequence representation, which factors in the location in the videos that a student visited. Then, we will present and evaluate CFA models based on this framework.

Definitions. Let \( v \in V \) denote video \( v \) in the set of videos \( V \) for a course, indexed chronologically (i.e., by release date of the videos). As in Section 2.4, let \( c \in O \) denote class \( c \) in the set of binary classes \( O = \{0, 1\} \), where \( c = 0 \) indicates a non-CFA submission and \( c = 1 \) is CFA. With \( u \in U \) as user \( u \) in the set of all users \( U \), we let \( U^v \subset U \) be the set of users who have a UV Pair for \( v \), and \( U^{v,c} \subset U^v \) be those who fall into class \( c \) with respect to their answer submission. For algorithm evaluation, we will generate training \( (U^v_T) \) and test \( (U^v_E) \) sets as subsets of \( U^v \); \( U^v_T \) and \( U^v_E \) are always chosen such that \( U^v_T \cap U^v_E = \emptyset \).

### 2.5.1 Position-Based Sequence Specification

We will divide each video into a number of intervals. Let \( h_v \) be the length (in sec) of \( v \). We define \( w_v \) to be the width that partitions \( v \) into \( N(w_v) = \lfloor h_v / w_v \rfloor \) uniform intervals, such that interval \( i \in \mathcal{P}^u(w_v) = \{1, ..., N(w_v)\} \) spans the range \([ (i-1) \cdot w_v, i \cdot w_v ] \). For each UV Pair, we can then model the video-watching behavior as a sequence of positions \( p^{u,v} = (\rho_1, \rho_2, ..., \rho_n, ...) \), where \( \rho_n \in \mathcal{P}^u(w_v) \) is the index of the \( n \)th position visited.

To generate these sequences, we first apply the same denoising procedure described in Section 2.3.2 to each event \( E_i \). Then, for each UV Pair, starting with \( p = () \) we do the following:

1. For \( E_1 \), append \( \lfloor p_1 / w_v \rfloor \) to \( p \).

---

\(^{16}\)Recall from Section 2.2.1 that we define a “video” to be all videos for a quiz.

\(^{17}\)For brevity, we will typically refer to \( p^{u,v} \) as just \( p \), with the understanding that it refers to the UV Pair in question.
2. Consider each sequential pair of events $E_i, E_{i+1}, i \geq 1$. If the state $s_i = \text{paused}$, then only $\lfloor p_{i+1}/w_v \rfloor$ is added to $p$. But if $s_i = \text{playing}$, then:

(a) If the event $e_i \neq \text{skip}$, then $(\lfloor p_i/w_v \rfloor + 1, \ldots, \lfloor p_{i+1}/w_v \rfloor - 1, \lfloor p_{i+1}/w_v \rfloor)$ is added to $p$.

(b) If $e_i = \text{skip}$, then $(\lfloor p_i/w_v \rfloor + 1, \ldots, \lfloor p'/w_v \rfloor - 1, \lfloor p'/w_v \rfloor, \lfloor p_{i+1}/w_v \rfloor)$ is appended instead.\footnote{Recall from Sec. 2.3.1 that when $E_i$ is a skip event, $p'_i$ is the position of the video player immediately before the skip.}

For example, suppose $h_v = 300$, $w_v = 15$, and a user generates $E_1 = \langle \text{play}, 0, 0, \text{playing}, 1.0 \rangle$, $E_2 = \langle \text{skip}, 200, 50, \text{playing}, 1.0 \rangle$, $E_3 = \langle \text{ratechange}, 230, 80, \text{playing}, 1.25 \rangle$, and $E_4 = \langle \text{pause}, 300, 127, \text{paused}, 1.25 \rangle$ on the video. Then, $p = (0, 1, 2, 3, 13, 14, 15, 16, \ldots, 20)$.

### 2.5.2 Model factors

There are (at least) three types of information for each $p^u_v$ that could have an effect on performance: positions, transitions, and time spent at positions.

**Positions**

First is the number of times a given position $i \in P_v(w_v)$ was visited. One would expect these to differ between CFA and non-CFA, because certain parts of videos will be more important to questions. We can see this by referring to the motifs that had correlations with increases or decreases in CFA probability. CFA sequences with revising motifs may have more visits to positions associated with the questions through repeating. On the other hand, non-CFA sequences with skimming motifs may have less visits to these important positions. Sequences with reflecting motifs may have more visits to important positions through pausing, too.
Transitions

Second is the number of transitions between the positions, i.e., the number of times a given tuple \((i, j)\) is a subsequence of \(p^{u,v}\). Considering each tuple \((\rho_n, \rho_{n+1})\):

- If \(\rho_{n+1} < \rho_n\), then the user had skipped back. We call this a backward transition.
- If \(\rho_{n+1} > \rho_n + 1\), then the user had skipped over the material in \((\rho_n, \rho_{n+1})\). This is a forward transition.
- If \(\rho_{n+1} = \rho_n + 1\), then the user moved directly to the next position. This is a direct transition.
- If \(\rho_{n+1} = \rho_n\), then the user had some event within the current position. This is a repeat transition.

We say that direct and repeat transitions are local, whereas backward and forward are non-local. As with positions, the transition factors can capture the motif behavior associated with changes in CFA probability, except in terms of sequences of visits, e.g., backward transitions capture \(S_b\) in a revising motif, and forward transitions capture \(S_f\) in a skimming motif.

Time Spent

Third is the amount of time spent at the different positions. One would expect these times to be indicative of CFA/non-CFA in a similar manner to visit frequencies.

We will consider four prediction models based on these factors. Three of them are likelihood-based: Discrete time Positions (DP), which incorporates the number of visits to each position; Discrete time Transitions (DT), which models transitions between positions; and Continuous time Transitions (CT), which factors in inter-arrival times between positions. For comparison, we include a standard SVM predictor that
uses position counts as features. In practice, the major advantage of the likelihood-based schemes over SVM is that its feature space is directly interpretable, leading e.g., to content analytics. Each model will be tested on each video separately, allowing us to compare results on a per-video basis for earliest detection (as opposed to the algorithms in Section 2.4 which trained over multiple videos).

2.5.3 Position-Based Modeling

Discrete Time Positions (DP)

For the DP model, video positions are treated as independent events. Let $f_{v,c} = [f_i]_{v,c} \in [0, 1]^{N(w_v)}$ be the probability distribution of visit frequency across positions $i \in P_v(w_v)$. This is estimated over the UV Pairs in the training set $U_T^{v,c}$ as

$$f_{v,c}^i = O_{v,c}^i / \sum_j O_{v,c}^j,$$  \hspace{1cm} (2.14)

where $O_{v,c}^i$ is the number of occurrences of $p_i$ over sequences in $U_T^{v,c}$. In other words, $O_{v,c}^i = \sum_{u \in U_T^{v,c}} O_{u,i}^{v,c}$, where $O_{u,i} = \sum_n \mathbb{I}_{\{\rho_n = i\}}$ is the number of times student $u$ was at $i$.

We test the ability of this model to identify which class each $u \in U_E$ belongs to. For this purpose, we compute the likelihood of observing $p$ on video $v$ to be in $c$, given $f^{v,c}$, as

$$L(p \mid f^{v,c}) = g^{v,c} \cdot \prod_n f_{\rho_n}^{v,c}.$$ \hspace{1cm} (2.15)

Then, the prediction $\hat{c} \in \{0, 1\}$ of the class for $p$ is determined by application of the Maximum a Posteriori (MAP) decision rule. But recall that there is a bias towards $c = 1$ for each course (see Table 2.1). As a result, we introduce a term $b_v \geq 0$ into MAP, which will be tuned through the cross validation procedure described in Section
\[ \hat{c} = \begin{cases} 
1 & g^{v,1}L(p | f^{v,1}) > g^{v,0}L(p | f^{v,0}) + b_v \\
0 & g^{v,1}L(p | f^{v,1}) < g^{v,0}L(p | f^{v,0}) + b_v \\
\mathbb{I}_{U \geq g^{v,0}} & \text{otherwise}
\end{cases} \tag{2.16} \]

where \( g^{v,c} = |\mathcal{U}^{v,c}_T|/|\mathcal{U}^c_T| \) is the estimated class bias for video \( v \), and \( U \) denotes a random number drawn from \([0, 1]\).

**Support Vector Machine (SM)**

Let \( O_T^v = [O_{u,i}]^v_T \) be the user-position matrix consisting of all \( u \in \mathcal{U}^v_T \), and let \( c_T^v \) be the vector of corresponding CFA scores. We fit \( \mathcal{M} : O_T^v \rightarrow c_T^v \) as an SVM over the training set, and then test the algorithm by comparing \( c_E^v \) to \( \mathcal{M}(O_E^v) \) for the users \( u \in \mathcal{U}_E^v \) in the test set. This function \( \mathcal{M} \) is described through the optimization in (2.8). In this case, we will use the standard linear kernel for \( \kappa \), and a standard regularization parameter \( C_v \) tuned through cross validation.

**2.5.4 Transition-Based Modeling**

In modeling transitions between positions, we will only consider one-step transitions. This is common in webpage clickstream analysis (e.g., [104]), and will be useful here since the state spaces we consider can be large, depending on \( u_v \).\(^{19}\)

**Aggregating Non-Local Transitions**

The cohort estimator for a Markov Chain model uses the fraction of transitions from state \( i \) to \( j \) in estimating the probability of transitioning from \( i \) to \( j \) [80]. We found this model not appropriate here, because the number of transitions between two non-

\(^{19}\)This may not be ideal because unlike sequences of webpages, learning builds on itself. It is harder to estimate higher order transitions due to position-specific data sparsity. We still see substantial benefit with a one-step model.
local positions is rather sparse, implying that there is not enough data to estimate these specific transitions.

To see this, we inspect the sequences $p^{v,c}$ for varying $w_v$. In particular, for each position in video $v$, we first find the total number of times each type of transition occurs, aggregated across the UV pairs. Then, we sum these totals over all positions, and find the fraction of each type of transition. We repeat this for each $w_v \in \{5, 10, \ldots, 600\}$ (i.e., through 10 min), and then average across the videos $v$ for each $w_v$. Figure 2.12 shows the result for each course, from which we make two observations for local and non-local transitions:

(i) Tradeoff between local transition types: As $w_v$ increases, the percentage of repeat transitions increases monotonically (from roughly 2% to 60% in each course), while the percentage of direct transitions decreases monotonically (from roughly 98% to 40% in each course). This is to be expected, since each position is increasing in size with $w_v$.

(ii) Infrequency of non-local transitions: The majority of transitions are local. For example, the largest fraction of backward transitions is just over 2% in ‘FMB’, occurring at $w_v = 120$.

As a result of the second observation, the models that follow will aggregate all observed forward transitions to form a single, uniform probability at each position, and likewise for backward transitions. To this end, we define $I_{i,k} = \{1, \ldots, i-1\}$ for $k = 1$; \{i\} for $k = 2$; \{i+1\} for $k = 3$; and \{i+2, ...\} for $k = 4$ to be the set of states constituting a backward ($k = 1$), repeat ($k = 2$), direct ($k = 3$), and forward ($k = 4$) transition at position $i$.

**Discrete Time Transitions (DT)**

In this model, we discretize time, discounting the interarrival times. Let $F^{v,c} = [f_{i,k}]_{v,c}^{(w_v)4} \in [0, 1]^{N(w_v)}$ be the matrix of transition probabilities, where $f_{i,k}^{v,c}$ is the prob-
Figure 2.12: Plot of the fraction of local (repeat and direct) and non-local (backward and forward) transitions for each window size $w_v$, averaged over all UV Pairs for each position and video $v$. The fraction of non-local transitions is very low.

ability that the next position will be in $I_{i,k}$ given the current is $i$. We also assume that the transitions are homogeneous, \textit{i.e.}, independent of time $n$.

Considering the sequences of positions $p$ across users $u \in U_{v,c}^T$, we obtain the number transitions from $i$ to $k$ as

$$O_{i,k}^{v,c} = \sum_{u \in U_{v,c}^T} \sum_{n} \mathbb{I}_{\{\rho_n=i, \rho_{n+1} \in I_{i,k}\}},$$  \hfill (2.17)
From (2.17), we estimate \( f_{i,k}^{v,c} = O_{i,k}^{v,c} / \sum_j O_{i,j}^{v,c} \), and the likelihood of \( p \) from user \( u \in U_v^c \) on video \( v \) is

\[
L(p | F^{v,c}) = f_{\rho_1}^{v,c} \cdot \prod_n f_{\rho_n,\rho_{n+1}}^{v,c},
\]

(2.18)

where \( f_{\rho_1}^{v,c} \) is the distribution at the initial position \( \rho_1 \) of \( p \), obtained from (2.14). The MAP decision rule for DT is the same as in (2.16), except with (2.18) in place of (2.15).

**Continuous Time Transitions (CT)**

This model incorporates the interarrival times between transitions. Rather than computing the time-varying transition probabilities, we instead work with the transition rates [80]. To this end, we define \( Q^{v,c} = [q_{i,k}]^{v,c} \in \mathcal{R}^{N(w_v) \times 4} \) as the transition rate matrix for the model, where \( q_{i,k}, k \neq 2 \) represents the rate of departure from position \( i \) and arrival at a position in \( I_{i,k} \).

Let \( r^{v,c} = [r_i]^{v,c} \in \mathcal{R}^{N(w_v)} \) be the vector of the total time spent by \( U_T^{v,c} \) in state \( i \). These terms are estimated as

\[
r_i^{v,c} = \sum_{u \in U_u^{v,c}} \sum_n 1_{\{\rho_n = i\}} \cdot d_n,
\]

(2.19)

where \( d_n \) is the duration of event \( n \) in \( p \) (see Section 2.3.1). In estimating the \( q_{i,k} \), we must also obtain the number of transitions from \( i \) to \( k \) over users \( u \in U_T^v \), i.e., the \( O_{i,k}^{v,c} \) from (2.17); with this, the \( q_{i,k}^{v,c} \) terms are estimated as

\[
q_{i,k}^{v,c} = \begin{cases} 
O_{i,k}^{v,c} / r_i^{v,c} & k \neq 2 \\
- \sum_{k \neq 2} q_{i,k}^{v,c} & k = 2
\end{cases}
\]

(2.20)
Finally, the likelihood of sequence $p$ for $u \in \Omega_v^E$ is computed:

$$L(p \mid Q^{v,c}) = \prod_{i,k: k \neq 2} (q_{i,k}^{v,c} o_{i,k} \exp(-q_{i,k}^{v,c} \cdot T_i)),$$  \hspace{1cm} (2.21)

where $o_{i,k} = \sum_n I\{\rho_n = i, \rho_{n+1} \in I_{ik}\}$, $k \neq 2$ is the number of transitions from $i$ to $k$ for the sequence $p$, and $T_i = \sum_n I\{\rho_n = i\} \cdot d_n$ is the time spent by $p$ in $i$. Once again, MAP is as in (2.16), except with (2.21) in place of (2.15).

We also considered another position-based model, Continuous Time Positions (CP), which used the time spent at each position in likelihood computation. We omit it because its results were strictly lower than these other three likelihood-based models.

### 2.5.5 Evaluation

In this section, we will investigate the following questions regarding these models, which are natural additions to the findings in Section 2.4:

1. **How beneficial is it to include video-watching positions and transitions for CFA prediction on individual videos?**

2. **How do the likelihood-based models compare against the SVM-based model?**

3. **Is one of position or transition-based model clearly better than the other, or would some combination be the best?**

4. **Is it beneficial to include position durations?**

**Skewed-Random (SR)**

To answer the first question, we will consider an algorithm that does not make use of clickstream data, Skewed-Random (SR). SR works like the naive predictor from Section 2.4.2, acting as a baseline for evaluating the gain from incorporating video-watching behavior; in particular, it finds the CFA bias $g^{v,1}$ over the training set $U_T^v$,.
and predicts \( \hat{c} = 1 \) of the time. It is important to note that the more sophisticated, collaborative filtering baselines from Section 2.4.2 that leverage similarities across users and/or quizzes are not applicable in this application of CFA prediction for individual videos.

**Metrics**

Let TP, FP, TN, and FN be the number of true and false positives, and true and false negatives obtained by a model on an evaluation set. As in Section 2.4.4 the first metric we consider is accuracy (Acc), i.e., \( \frac{TP + TN}{TP + FP + TN + FN} \). Since the quizzes are biased towards CFA (see Figure 2.4), we found that unconstrained maximization of accuracy during the tuning procedure (described below) led to high recall (rec), i.e., \( \frac{TP}{TP + FN} \) but low precision (prec), i.e., \( \frac{TP}{TP + FP} \). To avoid this, we will subject tuning to the constraint that the chosen parameters have at least 25% of the truly negative samples predicted negative, and likewise for the positives. To this end, the second metric we consider is the standard (balanced) \( F1 \) score, obtained as \( 2 \cdot \frac{\text{prec} \times \text{rec}}{\text{prec} + \text{rec}} \) [79]. As the harmonic mean of precision and recall, F1 is limited by the minimum of the two, capturing the tradeoff between them that is induced by this constraint.

Again, even a few percent improvement in these metrics can be a substantial benefit for CFA prediction. To see this more clearly, we can take an example of the first video in ‘FMB’, which is the earliest point of application of these algorithms in the harder of the two courses, and also the point at which the dropoff before the next video is the highest. Assuming that the total number of incorrect responses (roughly 1150) stayed the same, then for every 1% improvement in prediction accuracy, we could identify another 12 students who would get the question incorrect. Further, if the dropoff rate (roughly 25%) were to stay constant among the incorrect responses,
then each 1% improvement we give the chance to detect 3 more students that would otherwise drop off.

Training and Testing

For each algorithm and each video, we obtain the accuracy and F1-score metrics over $N$ evaluation iterations. In each iteration, we use the following procedure:

1. Divide the elements of $U_\nu$ into $K$ disjoint folds $U_1^\nu, U_2^\nu, \ldots, U_K^\nu$. In doing so, randomly allocate samples of CFA and non-CFA to folds, ensuring that the number of class instances is equal across folds (e.g., $|U_{k,c}^\nu| = |U_{l,c}^\nu| \forall k, l$).

2. Set $U_E^\nu = U_K^\nu$ and $U_T^\nu = U^\nu \setminus U_K^\nu$.

3. Tune the algorithm parameters $w_\nu$ and $b_\nu$ (for likelihood-based) or $C_\nu$ (for SVM) over the training set $U_T^\nu$, through the parameter tuning procedure described below.

4. With the tuned parameters, compute the features for each algorithm over $U_T^\nu$, and evaluate the fitted models on $U_E^\nu$.

The obtained metrics are averaged over the $N$ iterations. In our evaluation, we set $N = 10$ and $K = 5$.

Parameter Tuning

Each algorithm has two parameters that must be tuned. Let $W, B$, and $C$ be sets of potential values for the video width $w_\nu \in W$, the likelihood bias $b_\nu \in B$, and the regularization control $C_\nu \in C$. To tune these parameters for an algorithm, we apply a standard Cross-Validation (CV) procedure over the training set (similar to the one described in Section 2.4.4), which boils down to the following. First, for each CV iteration $k \in \{1, \ldots, K - 1\}$:
Table 2.5: Summary of the tuned parameters and quality metrics obtained across the videos for each course. $b_v$ applies to the likelihood-based algorithms, while $C_v$ is for SM. The average (avg) and standard deviation (s.d.) are taken first over the 10 evaluation sets for each video, and then over all the videos.

<table>
<thead>
<tr>
<th></th>
<th>$w_v$ (avg, s.d.)</th>
<th>$b_v$ ($C_v$) (avg, s.d.)</th>
<th>Acc (avg, s.d.)</th>
<th>F1 (avg, s.d.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>- -</td>
<td>- -</td>
<td>0.510 0.073</td>
<td>0.573 0.109</td>
</tr>
<tr>
<td>SM</td>
<td>62, 37</td>
<td>2.8E5, 4.0E5</td>
<td>0.545 0.064</td>
<td>0.583 0.138</td>
</tr>
<tr>
<td>DP</td>
<td>176, 116</td>
<td>4.9E-5, 1.3E-4</td>
<td>0.569 0.080</td>
<td>0.645 0.132</td>
</tr>
<tr>
<td>DT</td>
<td>263, 109</td>
<td>3.5E-5, 1.0E-4</td>
<td>0.572 0.084</td>
<td>0.614 0.165</td>
</tr>
<tr>
<td>CT</td>
<td>212, 99</td>
<td>2.1E-6, 3.7E-6</td>
<td>0.558 0.085</td>
<td>0.619 0.162</td>
</tr>
</tbody>
</table>

(a) ‘FMB’

<table>
<thead>
<tr>
<th></th>
<th>$w_v$ (avg, s.d.)</th>
<th>$b_v$ ($C_v$) (avg, s.d.)</th>
<th>Acc (avg, s.d.)</th>
<th>F1 (avg, s.d.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>- -</td>
<td>- -</td>
<td>0.531 0.069</td>
<td>0.607 0.108</td>
</tr>
<tr>
<td>SM</td>
<td>30, 15</td>
<td>1.8E5, 3.6E5</td>
<td>0.563 0.077</td>
<td>0.627 0.122</td>
</tr>
<tr>
<td>DP</td>
<td>75, 35</td>
<td>3.2E-4, 7.6E-4</td>
<td>0.589 0.093</td>
<td>0.654 0.176</td>
</tr>
<tr>
<td>DT</td>
<td>105, 72</td>
<td>3.7E-3, 7.8E-3</td>
<td>0.587 0.099</td>
<td>0.652 0.152</td>
</tr>
<tr>
<td>CT</td>
<td>71, 38</td>
<td>1.6E-5, 3.3E-5</td>
<td>0.587 0.097</td>
<td>0.661 0.165</td>
</tr>
</tbody>
</table>

(b) ‘NI’

1. Set $U_C^v = U_k^v$ and $U_R^v = U_T^v \setminus U_k^v$.

2. For each pair $(w_v, b_v) \in \mathcal{W} \times \mathcal{B}$ or $(w_v, C_v) \in \mathcal{W} \times \mathcal{C}$, find the result from training on $U_R^v$ and testing on $U_C^v$.

Then, average the accuracy values for each pair of parameters over the $K-1$ CV iterations. The pair which yields the highest average accuracy is selected, subject to the constraint described with the metrics above. We set $\mathcal{W} = \{5, 10, ..., 20, 30, ..., 600\}$, $\mathcal{B} = \{0, 2^{-60}, 2^{-58}, ..., 1\}$, and $\mathcal{C} = \{10^{-7}, 10^{-6.5}, ..., 10^7\}$; for each parameter, these choices ensured that most selections across videos did not lie on one of the endpoints.

**Results and Discussion**

Since there is a sharp decline in quiz submissions over time, we only consider those videos for which there are at least 100 samples of both CFA and non-CFA instances, so that there at least 20 samples from each group in each of the $K = 5$ folds. We
evaluate on the 24 videos for ‘FMB’ and the 32 for ‘NI’ that satisfy this criteria, which is a total of 56.

Summary information on the tuned $w_v$ and $b_v$ (or $C_v$) parameters, as well as the two performance metrics – Accuracy (Acc) and F1 – can be found for each algorithm and each course in Table 2.5. Here, we give the average (avg) and standard deviation (s.d.) of these values, taken across evaluation iterations for each video, and then across videos. The distribution of the performance values are plotted for each course in Figure 2.13; in each box, the performance on one video is one data point.

From Figure 2.13, we can see immediately that (i) the likelihood-based (DP, DT, and CT) algorithms perform substantially better than SR overall, and (ii) the SVM-based (SM) algorithm outperforms SR overall, but not as much as do the likelihood-based methods. Also, the improvement is higher for accuracy than it is for F1-score, which is expected since the tuning procedure monitors accuracy. In order to test for significance in the performance differences between each pair of models, we run a Wilcoxon Rank Sum (WRS) test for the null hypothesis that there is no difference between the distributions in Figure 2.13. The resulting $p$-values ($p$) from these tests are tabulated in Table 2.6 and verify our qualitative assessment from the boxplots.

We now answer the specific questions posed:

1: **Benefit of clickstream data.** We quantify how beneficial clickstream data can be for prediction. To do this, we compare the DP algorithm (which, from Table 2.5 and Figure 2.13, appears to have the highest overall quality) to the SR baseline.

Considering accuracy first, refer to Figure 2.13(a&c). DP is clearly shifted to the right relative to SR: for ‘FMB’, the shift in the mean of DP relative to SR is roughly 12%, and for ‘NI’, the improvement is roughly 11%. From Table 2.6, we see that this difference is also highly significant in both courses ($p$-value < 0.01). For F1-score, refer to Figure 2.13(b&d): again, we see that DP is shifted to the right relative to SR.
Figure 2.13: Boxplots of CFA prediction quality across both courses, considering accuracy and F1. Each datapoint is that measured on one of the videos considered. Qualitatively, we see that (i) the likelihood-based algorithms (DP, DT, and CT) outperform SR for both metrics, and (ii) the SVM-based (SM) algorithm also outperforms SR, but not as substantially.

overall, though not quite as substantially. The increase in means of roughly 13% for ‘FMB’ and 8% for ‘NI’ are both significant ($p$-value < 0.02 from Table 2.6).

For further analysis of the differences, in Figure 2.14 we show the comparison between DP and SR across the individual videos. The difference in the metrics obtained are shown for each video (specifically, DP minus SR). For accuracy in (a&c), we see that DP outperforms SR ($i.e.$, has a positive difference) for 98% of the videos.
Table 2.6: p-values from applying WRS tests to the boxplots in Figure 2.13.

<table>
<thead>
<tr>
<th></th>
<th>SR</th>
<th>SM</th>
<th>DP</th>
<th>DT</th>
<th>CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>–</td>
<td>0.013*</td>
<td>2.5E-3**</td>
<td>2.2E-3**</td>
<td>0.018*</td>
</tr>
<tr>
<td>SM</td>
<td>0.013*</td>
<td>–</td>
<td>0.25</td>
<td>0.20</td>
<td>0.63</td>
</tr>
<tr>
<td>DP</td>
<td>2.5E-3**</td>
<td>0.25</td>
<td>–</td>
<td>0.75</td>
<td>0.72</td>
</tr>
<tr>
<td>DT</td>
<td>2.2E-3**</td>
<td>0.20</td>
<td>0.75</td>
<td>–</td>
<td>0.28</td>
</tr>
<tr>
<td>CT</td>
<td>0.018*</td>
<td>0.63</td>
<td>0.72</td>
<td>0.28</td>
<td>–</td>
</tr>
</tbody>
</table>

(a) ‘FMB’, accuracy

<table>
<thead>
<tr>
<th></th>
<th>SR</th>
<th>SM</th>
<th>DP</th>
<th>DT</th>
<th>CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>–</td>
<td>0.36</td>
<td>0.014*</td>
<td>0.16</td>
<td>0.065.</td>
</tr>
<tr>
<td>SM</td>
<td>0.36</td>
<td>–</td>
<td>0.078.</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>DP</td>
<td>0.014*</td>
<td>0.078.</td>
<td>–</td>
<td>0.85</td>
<td>0.77</td>
</tr>
<tr>
<td>DT</td>
<td>0.16</td>
<td>0.15</td>
<td>0.85</td>
<td>–</td>
<td>0.98</td>
</tr>
<tr>
<td>CT</td>
<td>0.065.</td>
<td>0.15</td>
<td>0.77</td>
<td>0.98</td>
<td>–</td>
</tr>
</tbody>
</table>

(b) ‘FMB’, F1

<table>
<thead>
<tr>
<th></th>
<th>SR</th>
<th>SM</th>
<th>DP</th>
<th>DT</th>
<th>CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>–</td>
<td>0.092.</td>
<td>8.0E-3**</td>
<td>0.019*</td>
<td>0.015*</td>
</tr>
<tr>
<td>SM</td>
<td>0.092.</td>
<td>–</td>
<td>0.25</td>
<td>0.32</td>
<td>0.30</td>
</tr>
<tr>
<td>DP</td>
<td>8.0E-3**</td>
<td>0.25</td>
<td>–</td>
<td>0.91</td>
<td>0.79</td>
</tr>
<tr>
<td>DT</td>
<td>0.019*</td>
<td>0.32</td>
<td>0.91</td>
<td>–</td>
<td>0.94</td>
</tr>
<tr>
<td>CT</td>
<td>0.015*</td>
<td>0.30</td>
<td>0.79</td>
<td>0.94</td>
<td>–</td>
</tr>
</tbody>
</table>

(c) ‘NI’, accuracy

<table>
<thead>
<tr>
<th></th>
<th>SR</th>
<th>SM</th>
<th>DP</th>
<th>DT</th>
<th>CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>–</td>
<td>0.34</td>
<td>0.012*</td>
<td>0.045*</td>
<td>6.3E-3**</td>
</tr>
<tr>
<td>SM</td>
<td>0.34</td>
<td>–</td>
<td>0.052.</td>
<td>0.12</td>
<td>0.039*</td>
</tr>
<tr>
<td>DP</td>
<td>0.012*</td>
<td>0.052.</td>
<td>–</td>
<td>0.90</td>
<td>0.99</td>
</tr>
<tr>
<td>DT</td>
<td>0.045*</td>
<td>0.12</td>
<td>0.90</td>
<td>–</td>
<td>0.86</td>
</tr>
<tr>
<td>CT</td>
<td>6.3E-3**</td>
<td>0.039*</td>
<td>0.99</td>
<td>0.86</td>
<td>–</td>
</tr>
</tbody>
</table>

(d) ‘NI’, F1

across both datasets (all except one of the 56). For F1-score in (b&d), this drops to 88% (all except seven).

Note further that in Figure 2.13(b), the magnitudes on negative videos are substantially smaller than the magnitudes on the highest positive cases. In (d), however, two of the videos (3 and 27) have very high drops. One would expect that these would be instances where SR already had high performance due to a high bias (skew) in favor of either CFA or non-CFA (e.g., a video with an easy or a hard quiz). Surprisingly, the opposite is true: the CFA biases are close to 0.5 (roughly 0.46 in both
cases), and there are videos with smaller and larger biases for which DP outperforms SR substantially.

2: Likelihood vs. SVM. For this, we compare DP with SM. In Figure 2.13, we see that DP is shifted to the right for each course and metric. While the average improvements in accuracy of roughly 4% in both courses are not statistically significant ($p$-value > 0.1), the improvements in F1 of 11% for ‘FMB’ and 4% for ‘NI’ are significant ($p$-value < 0.1).

In Figure 2.14 we plot the difference between DP and SM for individual videos (as is done for DP and SR). The comparison here is consistent with the observations from the boxplots: the F1-score improves in 71% of the videos in (b&d), while the accuracy only improves in 66% of the cases in (a&c). Note also that, contrary to the comparison between DP and SR above, for each of the metrics and datasets, the videos in which DP has the highest gain over SM show significantly larger improvements than those in which SM shows the highest gains over DP. In other words, the improvement is less variable.

So, DP does outperform SM, but not as substantially as it outperforms SR. In fact, SM has a statistically significant gain over the baseline SR in terms of accuracy ($p$-value < 0.1 in Table 2.6).

3: Positions vs. transitions. For this, we compare DP to DT. In terms of accuracy, in Figure 2.13(a&c) we see that the algorithms are comparable for both courses. As for the F1-score in Figure 2.13(b&d), DP is modestly better on average, especially for ‘FMB’ where it has an improvement of roughly 5%. DT has a higher range in each case (excluding outliers), with generally lower performance than DP below quartile Q2 (e.g., in F1 for ‘FMB’) but, in accuracy for ‘FMB’, also higher above Q2. When considering individual videos DT and DP each perform better on roughly half of the videos in each course, with the exception of accuracy in ‘FMB’ for which DT has

20 We omit bar graphs over videos as in Figure 2.14 for brevity.
Figure 2.14: Difference in quality (in percent) between DP and SR (i.e., DP minus SR) and between DP and SM (DP minus SM) across individual videos, for each course and each metric. A positive bar indicates that DP performed better than the algorithm, while a negative bar indicates it performed worse. DP outperforms both algorithms in the majority of cases, but it outperforms SR in more cases than it does SM.
higher quality the majority of the time. Overall, the differences between DT and DP are not statistically significant for either course or metric ($p$-value $> 0.1$ in all cases).

4: Discrete vs. continuous. Finally, we compare DT to CT. In Figure 2.13, first consider accuracy. For ‘FMB’, DT is shifted to the right by roughly 3% relative to CT, whereas for ‘NI’, the algorithms are comparable. As to the F1-score, while DT and CT are comparable overall, the distribution for CT is slightly shifted to the right for both courses. Considering individual videos, DT outperforms CT on more videos for each dataset and metric. In particular, in Figure 2.14(a&c), it has higher accuracy in 71% of the cases, and higher F1 in 61% of the cases. Still, overall, the differences are not statistically significant for either course or metric ($p$-value $> 0.1$ in all cases).

2.5.6 Key Messages

Many aspects of position-based video behavior are useful for CFA prediction: the frequency of visits to each position (DP and SM), the frequency of transitions between positions (DT), and transitions incorporating holding times (CT). Each of these algorithms obtained higher quality than the SR baseline on both metrics and datasets tested, with statistically significant gains in most cases. Overall, the likelihood-based algorithms obtained the highest quality (with a slight edge given to DP), while the SVM-based algorithm forms a middle tier, and the SR baseline at the lowest.

The likelihood-based algorithms employ feature spaces that are representing user behavior directly; namely, positions visited and transitions. A significant advantage of this is that it leads to natural interpretations in terms of learner actions, which can be related to CFA scores for e.g., content analytics on an instructor dashboard through appropriate visualizations. Though the SVM-based model we tested is more complicated, and hence not as easily interpretable, it was still seen to obtain lower quality results than did the likelihood-based algorithms, i.e., the latter has better interpretability and better quality.
Also, in this evaluation, CFA prediction was done on a per-video basis. This underscores the applicability of these models to situations where there is not a lot of information across multiple lectures, e.g., for quick detection early in a course, or for short courses that have few videos to start with. As information across multiple videos becomes available, the algorithms evaluated in Section 2.4 will become effective too, and can be combined in an early detection prediction system.

2.6 Related Work

We discuss recent, key works on MOOC, student video-watching analysis, and student performance prediction.

2.6.1 MOOC Studies

With the proliferation of MOOC in recent years, there have been a number of analytical studies on these platforms. Some have focused on a more general analysis of all learning modes, e.g., [9, 67] studied learner engagement variation over time and across courses. Others have focused on specific modes, e.g., in terms of forums, we analyzed the decline in participation over 73 courses in [25]. There has also been work on identifying taxonomies of student motivation for enrolling in MOOCs, through e.g., designing and administering surveys [68] and interviews [115], and on studying how intention is predictive of course behavior. Our work on Learning Data Analytics presented in this chapter is fundamentally different from these works in that it (i) studies low level video-watching behavioral details, and (ii) explores the association between behavior with two modes: video and assessment. We will go into more detail on social learning in online courses in Chapter 3 too.
2.6.2 Video-Watching Analysis

Most existing works on learner video-watching behavior [65, 6, 73] have focused on session-level user characteristics (e.g., re-watching sessions) or aggregate quantities (e.g., number and duration of pauses). The work of [73] identified ways in which these types of quantities are correlated with student performance on quizzes, similar to our work in Section 2.2. Our work on motifs is fundamentally different than these because it represents behavior as sequences. The work in [94] is perhaps the most similar to ours in this regard, since it is also concerned with recurring patterns in clickstream sequences for MOOC users. The authors define a mapping of subsequences of events to predefined behavioral actions (e.g., skipping, slow watching) and perform approximate string search to locate these behaviors in clickstreams. Our work on motif extraction and identification differs from this in two important ways: (i) rather than assuming a predefined set of actions, we extract the recurring sequences directly using motif identification algorithms, and (ii) we are concerned with mapping motifs to assessment performance, in contrast to engagement.

2.6.3 Grade Prediction

Researchers have developed predictors for how students will perform on assessments (e.g., [71, 101, 14, 82, 86]) and for what their final grades will be (e.g., [77, 81, 90, 86]) in courses, some with application to traditional education settings and others for distance learning. Several techniques have been applied for this purpose, such as collaborative filtering algorithms [101, 14], support vector machines [90], and probabilistic graphical models [82, 81, 86]. Recently, [71] developed SPARFA-Trace, which traces a learner’s knowledge through the sequence of material accessed and questions answered. [77] proposed an algorithm to predict the final grade of each student after each assessment in a class based on the past history of students’ performance. Compared with these works, our research on prediction is unique in that (i) it focuses
on relating click-level data – video-watching behavior – to assessment performance, and (ii) it focuses on prediction for early detection. We emphasize that for the prediction on individual videos that was done in Section 2.5, the models used in each of these other works are not readily applicable to this setting, because similarities among users/quizzes is not available.

2.6.4 Webpage Clickstream Analysis

Webpage clickstream analysis \cite{104, 57, 96} remains an active area of research. Video-watching clickstreams are fundamentally different than these applications, which concern transitions between webpages rather than behavior within a single window.

2.7 Extensions

To summarize, in this chapter, we studied Learning Data Analytics (LDA) for video-watching behavior, quiz performance, and their association in MOOC. One clear extension of this work is to consider additional datasets, both MOOC and otherwise. We plan to investigate how other forms of behavioral data – like that from social learning networks – can be leveraged to enhance CFA prediction in these settings as well.

The models and algorithms we have proposed here can be extended for future work in several ways, too. For one, the event-based sequence representation in Section 2.3 can be generalized to optimize the selection of quantiles used divide the event lengths. Also, the position-based models in Section 2.5 can be extended to consider higher-order transitions and durations under a non-exponential assumption, to see whether the prediction quality can be improved further.

For the SVM scheme in Section 2.5 we believe that higher quality predictions could be obtained by passing these features through algorithms with higher com-
plexity (e.g., kernel-based SVM, rather than linear) to learn over higher dimensional spaces. Polynomial kernels were seen to work well for aggregate quantities in Section 2.4.4 for example. The downside is that this would eliminate interpretability entirely, which is important for providing instructors with behavioral analytics. Related to this, an interesting avenue of future work would be to use the position and transition matrices inferred over the CFA classes to generate recommendations guiding learner behavior to different locations in real time.

Additionally, due to the discriminative nature of the aggregate quantities in Section 2.2, a decision tree-type algorithm [82] that uses these quantities for branching may be able to enhance prediction quality further.

More generally, recall that students can have different motivations for taking MOOCs in the first place. In this chapter, we have limited our scope to those students who are interested in answering questions, which leaves an important step for future work to account for differing motivations in the behavioral analysis. The most objective way for this to be done is perhaps to release a questionnaire at the beginning of the course asking students to indicate their intentions for enrolling. Then, behavioral characteristics could be identified for each group separately, and the results compared.

Related to this is the fact that we have focused on video-watching behavior and in-video quiz performance. In accounting for different motivations, the definition of performance can be adjusted depending on what the student hopes to achieve, like high quiz scores, a broadened social network, or some combination of different metrics. Similarly, behaviors from different learning modes can be incorporated; in MOOC, this includes forum discussions, and in settings beyond MOOC, it can include behavior exhibited on any type of content integrated into the course.

Finally, we remark that the true test of the methods we have developed here is their impact on instructor interventions. We have shown that certain video-watching motifs are correlated with quiz scores, and quantified the quality of behavior-based
CFA prediction, but a larger question still remains: *How will an instructor make use of the motifs and predictions?* To investigate this, we are currently working with the learning technology company Zoomi Inc. [3] to integrate these algorithms into visualizations within an instructor dashboard that is being deployed to various learning scenarios. With these analytics in the hands of course instructors, we will determine their effect by monitoring overall changes in student performance based on the interventions that are made as a result.
Chapter 3

Social Learning Networks

A Social Learning Network (SLN) emerges when people exchange information on educational topics through structured interactions [21]. The proliferation of online communication has given rise to a number of SLN applications, ranging from Question and Answer (Q&A) sites (e.g., Quora), to enterprise social networks (e.g., Jive), to platforms for online education. They have created learning networks among askers/answerers, employees, and students, respectively.

SLN is a key learning mode in online courses, holding the promise of scaling up effective learning with rising enrollments (see Figure 3.1). To facilitate social learning, MOOC platforms typically provide discussion forums within each course, serving as the primary means for student-to-student and instructor-to-student interaction through series’ of user-generated posts and comments. These forums exhibit a few properties that raise into question how effective they are as they stand, though [69 87]. For one, they have sharp decline rates, with the amount of interaction dropping off rapidly soon after a course is launched, in line with the low completion rates of MOOCs discussed in Chapter 1. Within the discussions that persist, the forums will often exhibit information overload too [48], being flooded by discussions from many different students at once. This makes it infeasible for anyone to navigate the
Figure 3.1: The process of an instructor answering all student questions (left) does not scale with enrollment. In an SLN (right), the social component ideally allows the learning process to scale with the size of the network.

discussions to find course-relevant information, and is perhaps one of the reasons for the decline rates in the first place.

As a result, there is much room for improvement in the social learning process of online courses, which will in turn impact the quality of online learning more generally. The ability of today’s learning technology platforms to capture data about communication among students, in addition to the behavior they exhibit on other learning modes in the course, presents unique opportunities to do so.

As such, the focus of this chapter is two of our projects on the SLN of MOOC discussion forums [25, 20]. These works center around the following main question: How can SLN discussions be modeled, and how can these models be used to determine the efficiency of discussions? Developing such models lead to algorithms that can be used to automatically recommend communication among students that may be more effective. Additionally, like our work on Learning Data Analytics for video-watching behavior in the last chapter, this also leads to information that can be useful to course instructors, like visualizations of which students form communities of interaction within the larger SLN, and identification of which groups need assistance on which topics.
3.1 Objectives and Contribution

In this chapter, I will detail our methodology and results for the following three subquestions of the main question posed above:

Q1. How rapidly does the participation rate in the forums decline over time, and what behavioral factors maintain a healthy participation rate?

Q2. How can we model the ideal SLN in a given scenario, and how efficient is the observed information exchange relative to this ideal case?

Q3. What are the structural differences between the observed and ideal SLN, and how can they be used to recommend more efficient communication?

3.1.1 Modeling Thread Participation

To study Q1, we carried out an in-depth analysis to understand the factors that are associated with student participation in MOOC forums, which is the focus of Section 3.2. We first used regression models to understand what variables are significantly correlated with the number of posts (or users) on the forums in each day for each course. Some of the interesting findings were that (i) higher teaching staff participation in the discussions is associated with a higher discussion volume, but it does not slow down the decline rate, and (ii) quantitative and vocational courses attracted a smaller volume of discussion initially, but were also correlated with smaller decline rates. Along the way, we will also present some basic statistics about the MOOC provider Coursera (which is where our datasets are from), such as the total number of students that participate in the forums, the distribution on the number of posts for each student, and the distribution on thread lengths.

The statistical analysis makes it apparent that users have different needs in different stages of the course. In the first few days, the forum is often flooded with
“small-talk,” such as self-introductions. The primary goal in this stage should be to classify these threads and filter them out, at least for the benefit of those students who are looking for course-relevant information to help them through.

Then, beyond the first few days, the volume of small-talk often begins to drop. At this point, most of the threads are valuable, so it is important to ensure that they are visited by students (or instructors) that can provide answers to the questions raised there. More generally, the underlying efficacy of an SLN will hinge on the notion that strong social ties form between those seeking and those providing information on learning topics. This is the focus of Q2&3, which we turn to next.

### 3.1.2 Modeling SLN Efficiency

To study Q2&3, we started by proposing a novel framework for modeling SLN efficiency in Section 3.3, which compares the benefit obtained by users in the observed network to that which can be obtained through optimization. Each student is modeled individually as possessing certain levels of seeking (i.e., question asking) and disseminating (i.e., question answering) tendencies on a set of (latent) educational topics for the course. Additionally, our framework models the social structure of the SLN, which captures the level of connectivity between each pair of users. While there is some existing work on studying the content of MOOC forums (e.g., [55]) and some studying the graph structure (e.g., [56]), our work considers a unified view of both components.

Taken together, these components give us a natural way of defining user benefit, by comparing the match between a user’s seeking tendency and the disseminating tendencies of his/her neighbors (i.e., benefit from learning) and between the user’s disseminating tendencies and the seeking tendencies of his/her neighbors (i.e., benefit from teaching [43]). Ideally, these matches should be maximized. Therefore, our optimization searches for the SLN that is most compatible with the individual tendencies.
of the users, trading off the global utility (i.e., average benefit) and local utility (i.e., individual benefits). It also accounts for the fact that the amount a student will participate in the forums is constrained by his/her own resource limitations. Different from the optimization of users to questions proposed in [111], our method accounts for the difference between seeking and disseminating over a multidimensional topic space.

3.1.3 Evaluating SLN Efficiency

After developing our framework, we decided on several specific algorithms for inference and optimization. In particular, since our optimization involves several million variables corresponding to the weights in a (directed) user-to-user graph, it poses unique computational challenges. With the algorithms in place, we performed an efficiency evaluation on four MOOC datasets, presented in Section 3.4. In comparing the observed and optimal SLN, we found that SLN efficiency can be rather low (from 68% to 82% depending on the specific parameters and dataset), which indicates that much can be gained through optimization. We also found that the gains in global utility (i.e., average across users) can be obtained without making the distribution of local utilities (i.e., utility of individual users) less fair.

More generally, there are three sequential steps involved in improving the efficiency of an SLN:

- (i) defining the ideal SLN through an optimization,
- (ii) solving for the optimal SLN, and
- (iii) implementing the optimal SLN in practice.

In presenting and evaluating our efficiency framework, our focus in this chapter is the first two steps. The third will involve designing a system that enforces the optimal interaction structure in the corresponding SLN. We will briefly discuss possibilities
for this (e.g., a curated news feed for recommended interactions) at the end of the chapter too.

3.1.4 Summary of Contribution

Compared with related work (Section 3.5), this chapter makes the following main contributions to the field of Social Learning Networks:

- It gives the most comprehensive analysis of MOOC discussion forums to date – on 73 courses – and uses this dataset to identify new factors associated with student participation.

- It presents a novel framework for modeling SLN efficiency, capturing both the individual and social tendencies of learners in the network.

- It demonstrates that the efficiency of discussions in MOOC forums can be rather low, with room for improvement in global and individual utilities of the SLN simultaneously.

3.2 Decline Rate and Information Overload

Before presenting a statistical analysis of the decline rate and information overload, I will describe the dataset used here and how it was collected.

3.2.1 Forum Dataset

As stated, in this chapter, we focus on the discussion forums of MOOCs. A MOOC forum consists of a number of threads, with students able to create new threads or add content to existing threads. Each of these threads consists of one or more posts, sorted in chronological order. The first post is written by the person who created the
A MOOC discussion forum is a series of threads (left), with each thread having posts and comments (right).

A user may respond to a thread by adding a new post, or respond to a post within a thread by adding a new comment.

This structure is shown in Figure 3.2. Note that in our analysis, we will not distinguish between posts and comments, because there are only a small portion of comments in our dataset, and due to the user interface a student may be unaware of whether he/she is making a comment or adding a post.

Crawling and Parsing

We focused on all 78 courses that were available in the middle of July 2013 and that ended before August 10, 2013. Procedurally, we first manually calculated various properties for each course, such as the total video length and whether it was quantitative or vocational. Then, we crawled the forum content from Coursera’s server at a rate of 1 to 3 pages per second using Python and the Selenium library. Finally, we used Beautifulsoup to parse the html into text files.

Seven of these 78 courses became inaccessible while we were crawling or coding the data, and thus the 73 courses for which we have complete records were used for
## Table 3.1: List of course codes and their full names. All 78 were crawled and parsed.

<table>
<thead>
<tr>
<th>Code</th>
<th>Full Name</th>
<th>Code</th>
<th>Full Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>aboriginalworldviews-001*</td>
<td>Aboriginal World Views And Education</td>
<td>intropsych-001*</td>
<td>Introduction To Psychology</td>
</tr>
<tr>
<td>ADHD-001*</td>
<td>&quot;Pay Attention!!&quot; ADHD Through The Lifespan</td>
<td>introstats-001*</td>
<td>Statistics: Making Sense Of Data</td>
</tr>
<tr>
<td>analyticalchemistry-001*</td>
<td>Analytical Chemistry</td>
<td>latinasianculture-001</td>
<td>Latin American Culture</td>
</tr>
<tr>
<td>ancientgreeks-001*</td>
<td>The Ancient Greeks</td>
<td>lawandecon-001*</td>
<td>Property And Liability: An Introduction To Law And Economics</td>
</tr>
<tr>
<td>art-001*</td>
<td>Introduction To Art: Concepts &amp; Techniques</td>
<td>lead-e-001*</td>
<td>Inspiring Leadership Through Emotional Intelligence</td>
</tr>
<tr>
<td>audiomusicscengpart1-001*</td>
<td>Fundamentals Of Audio And Music Engineering</td>
<td>lyingcamera-001*</td>
<td>The Camera Never Lies</td>
</tr>
<tr>
<td>behavioralecon-001*</td>
<td>A Beginner’s Guide To Irrational Behavior</td>
<td>macroeconomics-001*</td>
<td>Principles Of Macroeconomics</td>
</tr>
<tr>
<td>biodiscovery-001*</td>
<td>BioDiscovery: A Quantitative Approach</td>
<td>malsoftware-001*</td>
<td>Malicious Software And Its Underground Economy</td>
</tr>
<tr>
<td>bluebrain-001*</td>
<td>Synapses, Neurons, And Brains</td>
<td>medicalneuro-001*</td>
<td>Medical Neuroscience</td>
</tr>
<tr>
<td>climatetiteracy-001*</td>
<td>Climate Literacy: Navigating Climate Change</td>
<td>mentalhealth-002*</td>
<td>Mental Health And Illness</td>
</tr>
<tr>
<td>compfinancc-004</td>
<td>Introduction To Computational Finance And Financial Ecmometrics</td>
<td>ml-003*</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>compilers-2012-002*</td>
<td>Compilers</td>
<td>modernpostmodem-001*</td>
<td>The Modern And The Postmodern</td>
</tr>
<tr>
<td>compilrs-003</td>
<td>Compilers</td>
<td>molevol-001*</td>
<td>Computational Molecular Evolution</td>
</tr>
<tr>
<td>comparstrategy-001</td>
<td>Competitive Strategy</td>
<td>mosfet-001*</td>
<td>Mos Transistors</td>
</tr>
<tr>
<td>crypto-007*</td>
<td>Cryptography 1</td>
<td>mythology-002*</td>
<td>Greek And Roman Mythology</td>
</tr>
<tr>
<td>datasci-001*</td>
<td>Introduction To Data Science</td>
<td>networkonline-001*</td>
<td>Social And Economic Networks: Models And Analysis</td>
</tr>
<tr>
<td>digitalmedia-001*</td>
<td>Creative Programming For Digital Media And Mobile App</td>
<td>neuralnets-2012-001*</td>
<td>Neural Networks For Machine Learning</td>
</tr>
<tr>
<td>drugdiscovery-001*</td>
<td>Drug Discovery, Development &amp; Commercialization</td>
<td>newwayhealthcare-001*</td>
<td>Interprofessional Healthcare Informatics</td>
</tr>
<tr>
<td>ecefoodandyou-001*</td>
<td>Economic Issues, Food, &amp; You</td>
<td>naturalprocessing-001*</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>einstein-001*</td>
<td>Understanding Einstein: The Special Theory Of Relativity</td>
<td>nutrition-002*</td>
<td>Nutrition For Health Promotion</td>
</tr>
<tr>
<td>epigenetics-001</td>
<td>Epigenetic Control Of Gene Expression</td>
<td>operations-002*</td>
<td>An Introduction To Operations Management</td>
</tr>
<tr>
<td>friendsmoneybytes-002*</td>
<td>Networks: Friends, Money, And Bytes</td>
<td>pgrm-003*</td>
<td>Probabilistic Graphical Models</td>
</tr>
<tr>
<td>gametheory-2012-002*</td>
<td>Game Theory</td>
<td>posa-001*</td>
<td>Pattern-Oriented Software Architectures For Concurrent And Networked Software</td>
</tr>
<tr>
<td>genschem1-001*</td>
<td>Chemistry: Conceptual Development And Application</td>
<td>projfun-002*</td>
<td>Functional Programming Principles In Scala</td>
</tr>
<tr>
<td>globalenergy-001*</td>
<td>Global Sustainable Energy: Past, Present And Future</td>
<td>programdesign-001*</td>
<td>Introduction To Systematic Program Design</td>
</tr>
<tr>
<td>g通报dcomp-001*</td>
<td>First-Year Composition 2.0</td>
<td>programming2-001*</td>
<td>Learn To Program: Crafting Quality Code</td>
</tr>
<tr>
<td>hci-003*</td>
<td>Human-Computer Interaction</td>
<td>rationing-001*</td>
<td>Rationing And Allocating Scarce Medical Resources</td>
</tr>
<tr>
<td>healthforall-002*</td>
<td>Health For All Through Primary Health Care</td>
<td>roac-001*</td>
<td>Cardiac Arrest, Hypothermia, And Resuscitation Science</td>
</tr>
<tr>
<td>healthinformatics-001*</td>
<td>Health Informatics In The Cloud</td>
<td>sciwrite-2012-001*</td>
<td>Writing In The Sciences</td>
</tr>
<tr>
<td>healthpolicy-002*</td>
<td>Health Policy And The Affordable Care</td>
<td>sdn-001*</td>
<td>Software Defined Networking</td>
</tr>
<tr>
<td>historyofrock1-001*</td>
<td>History Of Rock, Part One</td>
<td>secrets-001*</td>
<td>Archaeology’s Dirty Little Secrets</td>
</tr>
<tr>
<td>hollywood-001*</td>
<td>The Language Of Hollywood: Storytelling, Sound, And Color</td>
<td>socialPolitics-001*</td>
<td>Social Epidemiology</td>
</tr>
<tr>
<td>hswwinterface-001*</td>
<td>Image And Video Processing: From Mars To Hollywood With A Stop At The Hospital</td>
<td>sustainableag-001*</td>
<td>Sustainable Agricultural Land Management</td>
</tr>
<tr>
<td>images-2012-001*</td>
<td>Introduction To International Criminal Law</td>
<td>usefulgenetics-001*</td>
<td>Useful Genetics</td>
</tr>
<tr>
<td>intercriminallaw-001*</td>
<td>The Law Of The European Union: An Introduction</td>
<td>wealthofnations-001*</td>
<td>Generating The Wealth Of Nations</td>
</tr>
<tr>
<td>interoulaw-001*</td>
<td>Economic Issues, Food, &amp; You</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hwswinterface-001*</td>
<td>Image And Video Processing: From Mars To Hollywood With A Stop At The Hospital</td>
<td></td>
<td></td>
</tr>
<tr>
<td>introstats-001*</td>
<td>Statistics: Making Sense Of Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intrinsic-001*</td>
<td>Analytical Chemistry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>introductory-001*</td>
<td>Introduction To Art: Concepts &amp; Techniques</td>
<td></td>
<td></td>
</tr>
<tr>
<td>introstats-001*</td>
<td>Statistics: Making Sense Of Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>introstats-001*</td>
<td>Statistics: Making Sense Of Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>introstats-001*</td>
<td>Statistics: Making Sense Of Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>latinasianculture-001</td>
<td>Latin American Culture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lawandecon-001*</td>
<td>Property And Liability: An Introduction To Law And Economics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lead-e-001*</td>
<td>Inspiring Leadership Through Emotional Intelligence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lyingcamera-001*</td>
<td>The Camera Never Lies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>macroeconomics-001*</td>
<td>Principles Of Macroeconomics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>malsoftware-001*</td>
<td>Malicious Software And Its Underground Economy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>medicalneuro-001*</td>
<td>Medical Neuroscience</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mentalhealth-002*</td>
<td>Mental Health And Illness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ml-003*</td>
<td>Machine Learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>modernpostmodem-001*</td>
<td>The Modern And The Postmodern</td>
<td></td>
<td></td>
</tr>
<tr>
<td>molevol-001*</td>
<td>Computational Molecular Evolution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mosfet-001*</td>
<td>Mos Transistors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mythology-002*</td>
<td>Greek And Roman Mythology</td>
<td></td>
<td></td>
</tr>
<tr>
<td>networkonline-001*</td>
<td>Social And Economic Networks: Models And Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>neuralnets-2012-001*</td>
<td>Neural Networks For Machine Learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>newwayhealthcare-001*</td>
<td>Interprofessional Healthcare Informatics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>naturalprocessing-001*</td>
<td>Natural Language Processing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nutrition-002*</td>
<td>Nutrition For Health Promotion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oldglobe-001*</td>
<td>Growing Old Around The Globe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>operations-002*</td>
<td>An Introduction To Operations Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>orgchem2a-001*</td>
<td>Intermediate Organic Chemistry - Part I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pgrm-003*</td>
<td>Probabilistic Graphical Models</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posa-001*</td>
<td>Pattern-Oriented Software Architectures For Concurrent And Networked Software</td>
<td></td>
<td></td>
</tr>
<tr>
<td>projfun-002*</td>
<td>Functional Programming Principles In Scala</td>
<td></td>
<td></td>
</tr>
<tr>
<td>programdesign-001*</td>
<td>Introduction To Systematic Program Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>programming1-2012-001*</td>
<td>Learning To Program: The Fundamentals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>programming2-001*</td>
<td>Learn To Program: Crafting Quality Code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rationing-001*</td>
<td>Rationing And Allocating Scarce Medical Resources</td>
<td></td>
<td></td>
</tr>
<tr>
<td>roac-001*</td>
<td>Cardiac Arrest, Hypothermia, And Resuscitation Science</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sciwrite-2012-001*</td>
<td>Writing In The Sciences</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sdn-001*</td>
<td>Software Defined Networking</td>
<td></td>
<td></td>
</tr>
<tr>
<td>secrets-001*</td>
<td>Archaeology’s Dirty Little Secrets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>socialPolitics-001*</td>
<td>Social Epidemiology</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sustainableag-001*</td>
<td>Sustainable Agricultural Land Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>usefulgenetics-001*</td>
<td>Useful Genetics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wealthofnations-001*</td>
<td>Generating The Wealth Of Nations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: List of course codes and their full names. All 78 were crawled and parsed. A * indicates a course that was used in the statistical analysis (73/78), and a △ indicates one that passed the Shapiro test (51/78, see Section 3.2.2).
statistical analysis. Table 3.1 gives the entire list of courses. In total, our dataset consists of approximately 830K posts (Section 3.2.2 presents more details).

**Categorizing the Courses**

For the purpose of comparison across course types, we categorize a course as quantitative or non-quantitative, and vocational or non-vocational. If a substantial portion of a course requires the students to carry out mathematical or statistical analysis, or to write computer programs, then we consider it to be quantitative. If a course’s material is directly relevant to jobs that require high school or college degrees, or it is plausible to see the course offered in a typical university’s continuing education division, then we consider it to be vocational. For summary purposes, we use these categories to partition the data into three groups: a course could be (i) vocational, (ii) science or applied science (i.e., quantitative but not vocational), or (iii) humanities and social sciences (i.e., neither quantitative nor vocational).

**Small-Talk vs. Course-Related Discussions**

Now, discussion threads in any MOOC can be roughly categorized into the following three groups:

- *Small-talk* conversations that are not course-related, such as a self-introductions or requests to form study groups.

- *Course logistics* such as when to submit homework, how to download lecture videos, and so on.

- *Course-specific discussions* that can range in scope from specific to open-ended.

The last two groups can be referred to collectively as course-related discussions. A similar taxonomy distinguishing between conversational (i.e., small-talk) and informational (i.e., course-related) discussions is given in [59].
We wonder, what is the fraction of small-talk relative to course-related discussion in MOOC? To investigate this, we randomly chose 30 threads from each course and hired workers from Amazon Mechanical Turk (AMT) to label them. Each thread was labeled by 3 workers; we used a majority vote among them to assign the final labels.

Analyzing the results, we found that a substantial portion of the course forums had more than 10% small-talk, and that those in the humanities and social sciences category tended to have a higher portion of small-talk than the others. We also studied how the portion of small-talk changed over time; since it is infeasible to label a significant portion of the threads, we used a Support Vector Machine (SVM) to classify the threads, using the labels from AMT as training data. We saw that at the beginning of the courses, the percentage of small-talk was high across different categories, and then it dropped over time. However, for humanities and social sciences courses, on average more than 30% of the threads were classified as small-talk even long after the start dates.

Both small-talk and course-related discussions are important to the forum experience on MOOC. At the same time, we can identify a number of scenarios in which a student would prefer to read course-related threads only, such as when he/she is reviewing for an exam or searching for help with a homework problem. Hence, it is important to be able to separate conversational from informational discussions in MOOC, in order to assist user navigation as needed, especially when the number of new threads is already excessive. It is worth mentioning that we have designed and evaluated an algorithm for ranking threads by relevance that was seen to have superior quality to standard ranking schemes. For the detailed methodology, specification, and evaluation, see [25].
### 3.2.2 Analysis of Forum Activity Decline

With a basic understanding of the datasets, we will now investigate the relationship between certain factors and the forum activity decline rate. Our dependent variables are the number of posts on the $t$th day in the $i$th course ($y_{i,t}$) and the number of distinct users that participate in discussion on the $t$th day in the $i$th course ($z_{i,t}$). Both of these are important to social learning in MOOCs.

We define the following eight factors that could be relevant to $y_{i,t}$ and $z_{i,t}$ (in a similar vein to how we defined aggregate quantities in Section 2.2):

1. **Quantitative** ($Q_i$): An indicator variable that is 1 if the course is quantitative, and 0 if it is not.

2. **Vocational** ($V_i$): An indicator variable that is 1 if the course is vocational.

3. **Video length** ($L_i$): The sum of the length (in hours) of all lecture videos in the course.

4. **Course duration** ($D_i$): The total length (in weeks) of the course.

5. **Peer-grading** ($P_i$): An indicator variable that sets to 1 if at least one assignment in the course is reviewed/graded by peer students.

6. **Graded homework** ($H_i$): The total number of homework assignments that are graded by the teaching staff in the course.

7. **Staff activity** ($S_i$): The number of posts the teaching staff makes throughout the course.

8. **Intrinsic popularity** ($M_i$ or $M'_i$): The volume of forum discussion in the beginning of the course. If the dependent variable is the number of posts $y_{i,t}$, this is defined as $M_i$, the median number of posts in the first three days of the $i$-th course. On the other hand, if it is the number of users $z_{i,t}$, then this is defined as $M'_i$, the number of distinct users in the first three days. Roughly speaking, this variable captures the “intrinsic popularity” of each course, e.g., it is likely that a course on public speaking will attract higher enrollment than a course on advanced machine learning.
Dataset Statistics

We will now examine the key statistics of 73 courses (see Table 3.1) in terms of the factors defined above.

Course categories (1&2). Among the 73 courses, 37 of them were considered quantitative and 8 of them vocational. There are 6 courses that were both quantitative and vocational.

Video length (3). The mean video length across the courses is 12.71 hours (standard deviation (s.d.) = 7.85). We further analyzed the breakdown of video length by categories, and did not see discrepancies between them.

Course length (4). All courses in our dataset are between 4 and 14 weeks long. The mean length is 59 days (s.d. = 15.3).

Homework assignments (5&6). The mean number of staff-graded assignments per course is 10 (s.d. = 10.88). Out of the 73 courses, 6 of them did not have any staff-graded assignments. A total of 39 courses had peer-graded homework; among these, 5 were vocational courses, 11 were science or applied science, and 23 were humanities and social science.

Staff activity (7). On average, there were 366.9 posts in each course made by the teaching staff (s.d. = 446.1). Two courses had no staff posts.

Students in the Forums

Overall, there are 171,197 threads, 831,576 posts, and 115,922 distinct users in our dataset. Figure 3.3(a) shows the distribution of the number of posts each student made (in log-log scale). Due to the relatively large variation in participation by user, we classify each student as an “active” or “inactive” forum user, considering a student active if he/she made at least 2 posts in a course and inactive otherwise.

\footnote{Choosing 2 as the threshold is rather arbitrary. The goal here is to show that many students make a small number of posts.}
Figure 3.3: Distributions of (a) student posts and (b) decline rates. Many students make only a few posts, and many courses have sharp decline rates.

Figure 3.4 shows the distribution of the number of students and active students in different courses, separated by category. The reduction is substantial: while the average number of students per course is 1835 (s.d. = 1975), the average for active students is only 1070 (s.d. = 1218).

Decline Rates

We will now analyze how the dependent variables in the regression – the number of posts \( y_{i,t} \) and the number of users \( z_{i,t} \) – change over time for different courses. Figure 3.5(a) shows the variables \( \{y_{i,t}\}_{t \geq 1} \) for four randomly selected courses. Each course presented here exhibits a decline in participation over time. The rest behave similarly as well, as can be seen in Figure 3.3(b), where the distribution of the decline rate (defined as the slope of a linear regression model) over all the courses is shown (mean = \(-5.0\) and s.d. = \(8.7\); 72 of 73 are negative\(^2\)). The variables \( \{z_{i,t}\}_{t \geq 1} \) are qualitatively similar.

\(^2\)The only course with a positive rate also has a negative rate when we count the number of distinct users instead.
Figure 3.4: Distribution of the number of students (left) and active students (right) per course, by category.

We also study the distribution of the differences in post counts between two consecutive days in the same course. It is not uncommon to see outliers due to large fluctuations in discussion, especially near the start date or when homework/exams are due. After we remove the top and bottom 3% outliers from each course, we find that the count differences do follow a Gaussian distribution in most cases, though. 51/73 of the courses pass the Shapiro-Wilk test for normality (see Table 3.1), with p-values ≥ 0.01. The corresponding Q-Q plots for the four randomly chosen courses can be seen in Figure 3.5(b); each of these four passed the test.

**Postulation of a Model**

Based on this, we postulate the following linear model for the post counts.

Let \( y_{i,t} \) be the number of posts on the \( t \)th day in the \( i \)th course. We assume \( y_{i,t+1} - y_{i,t} \sim \mathcal{N}(\mu_i, \sigma_i) \), i.e., \( y_{i,t} = \sum_{j \leq t} \mathcal{N}(\mu_i, \sigma_i) = \mathcal{N}(t \mu_i, \sqrt{t} \sigma_i) \). Here, the mean term grows linear in \( t \) while the “noise term” grows linear in \( \sqrt{t} \). When \( t \) is sufficiently large, the mean term dominates \( y_{i,t} \).

In other words, we may model \( y_{i,t} = A_i t + B_i + \epsilon_{i,t} \), where \( A_i \) and \( B_i \) only depend on the factors of the \( i \)th course. While serial dependency may be present, we believe
Figure 3.5: The decline of forum activities over time for 5 of the 73 courses, chosen randomly. In (a), the number of posts by day is plotted, and a regression line is added. In (b), Q-Q plots for these courses of the difference in count between two consecutive days are plotted, after removing 6% outliers.

this factor-adjusted, deterministic, linear trend is sufficient to explain the pattern; this is confirmed by our subsequent empirical results.

**Regression Model**

We model the number of posts $y_{i,t}$ to be linearly related to (i) the course factors, (ii) the variable $t$, and (iii) the interacting terms between $t$ and the factors, as

$$y_{i,t} = At + B,$$

where

$$A = \beta_1Q_i + \beta_2V_i + \beta_3L_i + \beta_4D_i + \beta_5P_i + \beta_6S_i + \beta_7H_i + \beta_8M_i + \beta_{16}$$
and
\[ B = \beta_0 + \beta_9 Q_i + \beta_{10} V_i + \beta_{11} L_i + \beta_{12} D_i + \beta_{13} P_i + \beta_{14} S_i + \beta_{15} M_i. \]

We can therefore view the parameters \( \beta_1, \ldots, \beta_8 \), and \( \beta_{16} \) as being related to the participation decline rate, and \( \beta_0, \beta_9, \ldots, \beta_{15} \) as to the initial participation volume.

We fit these parameters to our dataset using ordinary least-squares regression. The results are shown in Table 3.2. For the fit to \( y_{i,t} \), we make several observations, including the following:

**Intrinsic popularity.** It is evident that there is a significant relationship between the number of posts \( y_{i,t} \) and the intrinsic popularity \( M_i \). On the other hand, any impact of \( M_i \) on the decline rate in the long run appears very light.

**Course categories.** The coefficients \( Q_i, V_i, Q_t, \) and \( V_t \) for quantitative and vocational courses suggest that while they are associated with a lower volume of forum participation initially, in the long run they tend to experience lower decline rates (all \( p \)-values \( \leq 10^{-6} \)).

**Teaching staff.** The results for the number of teaching staff posts \( S_i \) are surprising: while teaching staff active participation in the forum is associated with a higher volume of discussion (for every additional post by the teaching staff, there are on average 6.05 additional posts in the forum each day), in the long run it does not seem to reduce the decline rate. In fact, there is evidence that an increase in staff participation is correlated with a higher decline over time (\( p \)-value = 0.021).

**Peer-reviewed homeworks.** The relationship between \( y_{i,t} \) and the number of peer-reviewed homeworks \( P_i \) is similar: while the presence of peer-reviewed homework is associated with 88.29 additional posts per day on average, it is also correlated with a higher decline rate (\( p \)-value = 0.018).

We remark that the overall \( p \)-value of the model is \( 2.2 \times 10^{-16} \), suggesting overall significance. Also, we diagnosed the residuals, which did not seem to elicit any heteroscedastic pattern. We further checked the differences between the slope of \( t \) (i.e.,
the quantity \( A_i \) in the regression) under our proposed multivariate regression model and the counterpart for univariate regression with a single parameter imposed on \( t \) for each course \((i.e., \) the slopes computed as in Figure 3.5(a)); these differences were reasonably small in magnitude.

The results for \( z_{i,t} \) are qualitatively similar: quantitative and vocational courses \((Q \) and \( V)\) are associated with a smaller volume of distinct users on the forums initially, but also with smaller decline rates over time \( (all \ with \ p\text{-values} \leq 10^{-6}); \) teaching staff participation \((S)\) is associated with an increased number of distinct users on the forums \((p\text{-value} = 0.00994), \) but also with higher decline rates \((p\text{-value} = 0.038); \) and the presence of peer-grading \((P)\) is associated with an increase in the total number

<table>
<thead>
<tr>
<th></th>
<th>On ( y_{i,t} )</th>
<th>On ( z_{i,t} )</th>
<th>On ( \log(z_{i,t}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>18.276</td>
<td>70.252**</td>
<td>4.268**</td>
</tr>
<tr>
<td>( Q_i )</td>
<td>1.511**</td>
<td>0.847**</td>
<td>0.014**</td>
</tr>
<tr>
<td>( V_i )</td>
<td>3.328**</td>
<td>1.463**</td>
<td>0.011*</td>
</tr>
<tr>
<td>( L_i )</td>
<td>-0.071**</td>
<td>-0.024**</td>
<td>1.20**</td>
</tr>
<tr>
<td>( D_i )</td>
<td>0.034**</td>
<td>0.025**</td>
<td>0.001**</td>
</tr>
<tr>
<td>( P_i )</td>
<td>-0.631*</td>
<td>-0.375**</td>
<td>0.003</td>
</tr>
<tr>
<td>( S_i )</td>
<td>-0.168*</td>
<td>-0.067*</td>
<td>-0.001*</td>
</tr>
<tr>
<td>( H_i )</td>
<td>0.000</td>
<td>-0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>( M_i ) or ( M'_i )</td>
<td>-0.007**</td>
<td>-0.005**</td>
<td>0.000*</td>
</tr>
<tr>
<td>( Q_i )</td>
<td>-13.975</td>
<td>-23.737**</td>
<td>-0.185*</td>
</tr>
<tr>
<td>( V_i )</td>
<td>-135.567**</td>
<td>-61.404**</td>
<td>-0.153</td>
</tr>
<tr>
<td>( L_i )</td>
<td>1.960*</td>
<td>-0.049</td>
<td>1.36*</td>
</tr>
<tr>
<td>( D_i )</td>
<td>-0.561</td>
<td>-0.624**</td>
<td>-0.010**</td>
</tr>
<tr>
<td>( P_i )</td>
<td>88.289**</td>
<td>32.005**</td>
<td>0.247**</td>
</tr>
<tr>
<td>( S_i )</td>
<td>6.050*</td>
<td>3.249**</td>
<td>0.074**</td>
</tr>
<tr>
<td>( H_i )</td>
<td>1.398*</td>
<td>0.973**</td>
<td>2.79*</td>
</tr>
<tr>
<td>( M_i ) or ( M'_i )</td>
<td>0.481**</td>
<td>0.360**</td>
<td>0.003**</td>
</tr>
<tr>
<td>( t )</td>
<td>-1.864**</td>
<td>-1.980**</td>
<td>-0.071*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0.555</th>
<th>0.467</th>
<th>0.530</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adj. ( R^2 )</td>
<td>0.554</td>
<td>0.465</td>
<td>0.526</td>
</tr>
<tr>
<td>Num. obs.</td>
<td>5074</td>
<td>5074</td>
<td>1711</td>
</tr>
</tbody>
</table>

Table 3.2: Regression parameters for forum activity models, with levels of significance indicated for each. ** indicates \( p < 0.01, * \) indicates \( p < 0.05, \) and . indicates \( p < 0.1. \)
of distinct users (\(p\)-value = \(5.94 \times 10^{-10}\)), but also with higher decline rates (\(p\)-value = 0.0016). The \(p\)-value of this model is \(2.2 \times 10^{-16}\), which again suggests overall significance, and the residuals do not show any obvious patterns here either.

**More Robust Linear Model**

While these linear regression models are in general robust against noise, we also performed analysis on the subset of courses whose residuals exhibited normality (the ones with “nice” behavior) to see whether or not the conclusions were consistent. Specifically, we chose the 51 courses whose count difference in posts passed the Shapiro test with \(p\)-value \(\geq 0.01\) after removing the top and bottom 3\% outliers (see Table 3.1). We used the same format and regressors as the previous model, except we fit the data to the logarithm of the number of distinct users \(\log(z_{i,t})\); this transformation resulted in even higher model significance.

The last column of Table 3.2 presents the results for this. Since the variables for video length and graded homework (\(L_i\), \(L_{it}\), \(H_i\), and \(H_{it}\)) are not statistically significant, we remove them. The conclusions we see here are mostly consistent with those made previously, the exception being the terms involving the staff participation \(S_i\): while staff participation is still associated with an increased number of distinct users each day on average, the correlation with the decline rate is negligible (coefficient = \(-1.47 \times 10^{-3}\), \(p\)-value = 0.04).

We performed further analysis to test the residuals for normality. The \(p\)-value of the Shapiro test is 0.148, which indicates that our model fits well for these 51 courses.

### 3.2.3 Analyzing Information Overflow

Information overflow can be difficult to quantify. We investigate its presence in MOOC forums by posing the following question: *will the creation of more threads concurrently reduce the average attention that each receives on average?*
Figure 3.6: Distribution of thread lengths over the 73 courses used for statistical analysis. The average thread has about 5 posts, with a heavy tail of a few threads receiving hundreds.

There are many potential ways of quantifying “attention,” perhaps without a single “right” answer. Similar to [98, 13], we will use the number of posts in a thread, which we will consider to be its length, as a proxy for attention. Other possible metrics include the total number of views or votes on each thread, and ideally could be combined with posts, but such information is not publicly available on Coursera: the number of views on each page is only available to course instructors, and the forums only present the difference between the number of up-votes and down-votes for each post, rather than the number of each (see Figure 3.2).

**Diluted Thread Attention**

We briefly present the distribution of the thread lengths, our measure of attention, in the forums. Figure 3.6(a) gives the distribution in log-log scale over all 73 courses; the mean is 4.98 (median = 2 and standard deviation = 8.65), with a heavy tail of a few threads receiving hundreds of posts. Figure 3.6(b) gives boxplots of the thread lengths by course category; threads in the social-sciences courses tend to have smaller lengths.
We wonder whether having more newly created threads in a given time period will dilute the attention that each of the threads receives. Let \( h_j \) refer to a discussion thread within a given course, and let \( \ell_j \) be the length of \( h_j \) as defined previously. Also, let \( f(h_j, t) \) be the total number of other threads in this course that were created within the window of \( t \) days before and after \( h_j \) was created; for example, if \( h_j \) was created on July 2 at 3pm, then \( f(h_j, 1) \) is the number of threads besides \( h_j \) that were created between July 1 at 3pm and July 3 at 3pm.

Figure 3.7(a) shows the plot of \( \ell_j \) against \( f(h_j, 1) \) for all threads in our dataset. Visually, there is strong evidence that threads receive less attention when there are more created in the same window. To quantify this, we first attempted to fit this with a linear model, but found its explanatory power quite low (\( R^2 \) is below 0.02). As a result, we instead resorted to two-sample statistical testing, partitioning the threads into two groups. The first, \( G_1 \), contains all the threads \( h_i \) such that \( f(h_i, 1) \leq 140 \), and the second, \( G_2 \), contains the rest. The threshold number 140 is chosen so that the size and variances were within a factor of two between the groups (size of \( G_1 \) is 44,971, \( \text{Var}_{h_i \in G_1}(\ell_i) = 103.94 \); size of \( G_2 \) is 76,890, \( \text{Var}_{h_i \in G_2}(\ell_i) = 62.14 \)). Referring to \( G_1 \) as the small group and \( G_2 \) as the large group, Figure 3.7(b) gives the boxplot of the log of thread lengths in both groups.

We now test the null hypothesis is that the small group’s thread length is no greater than the large group’s thread length against the alternative hypothesis that the small group’s thread length is greater than the large group’s thread length. A \( t \)-test yielded a \( t \)-statistic of 40.3 and a \( p \)-value \( \leq 2.2 \times 10^{-16} \). We also carried out a Wilcoxon Rank Sum (WRS) test, which yielded a similar \( p \)-value \( \leq 2.2 \times 10^{-16} \). These tests indicate that we can reject the null hypothesis with high confidence with respect to the mean and median, respectively. Therefore, there is strong evidence that the creation of more threads concurrently is correlated with a reduction in the
Figure 3.7: Discussion thread length against the number of threads created around the same time, using a window size of 1. There is strong evidence that the creation of more threads in an interval of time is negative correlated with the attention each receives.

attention that each thread receives, which is evidence that information overload in the forums indeed does exist.

### 3.2.4 Key Messages

Our statistical analysis over 73 Coursera course forums yields a few main messages:

- Several course factors, like whether it is quantitative and/or vocational, the amount of staff participation, and the number of peer-reviewed homeworks, are either positively or negatively correlated with course participation over time, sometimes in surprising ways.

- Information overload persists in MOOC forums, causing each thread to receive a smaller amount of attention (quantified as posts) when more are created.

We turn next to quantifying how efficient the communication between learners is on MOOC. This will help us discover whether a more optimal state that benefits the
SLN as a whole, as well as the individuals that it is comprised of, can be realized, with the hope that it will improve the dropoff rate and the quality of online learning in general.

3.3 Modeling SLN Efficiency

To evaluate the efficiency of a social learning network, we pose the following question: *How much are users benefiting from the observed network structure relative to how much they could benefit from an optimized structure?* In this section, we will present our efficiency framework, consisting of our graph model (Section 3.3.1), utility model (Section 3.3.2), and optimization (Section 3.3.3).

3.3.1 Graph Modeling of SLN (W, S, and D)

We will first define and model the essential components of an SLN.

**Users**

At its core, an SLN is a network of users *(i.e., learners)* sharing information on different topics. Let \( u \in \mathcal{U} \) denote user \( u \) in the set of users \( \mathcal{U} = \{1, 2, \ldots \} \) that comprise the SLN. In Section 3.2, our focus was on user participation over time.

**Network**

In studying efficiency, we are interested in the interaction structure between users. We define \( W = [w_{u,v}] \), for \( u, v \in \mathcal{U} \) \((w_{u,u} = 0)\) to be the weighted adjacency matrix of the user-to-user network, where \( w_{u,v} \) represents the spread of information from \( u \) to \( v \). More concretely, we consider \( 0 \leq w_{u,v} \leq 1 \) to be the probability that \( u \) will respond to \( v \) when \( v \) makes a post, with \( w_{u,v} \neq w_{v,u} \) in general.
Topics

Discussions in an SLN center around a series of (possibly latent) topics. Let $k \in \mathcal{K} = \{1, 2, \ldots\}$ denote topic $k$ in the set $\mathcal{K}$ for the course.

Seeking and Disseminating

With respect to each topic, a user will have some tendency towards disseminating information (i.e., providing answers or facts about the material) or seeking information (i.e., asking questions about material). In order to capture this behavior, we define $s_{u,k}$ to be user $u$’s seeking tendency on topic $k$, and $d_{u,k}$ as his/her disseminating tendency on $k$, with $S = [s_{u,k}]$ and $D = [d_{u,k}]$. In an ideal scenario, users seeking information would be receiving responses from those disseminating on the same subject matter, i.e., the corresponding $w_{u,v}$ would be large.

3.3.2 Utility Modeling of SLN (B and E)

We identify two types of user benefit from an SLN: learning benefit and teaching benefit.

(1) Learning Benefit

Intuitively, user $u$ will gain from having higher connections to those who tend to disseminate information on topics that $u$ asks questions on. We quantify this as $s_{u,k} \cdot f(\sum_v w_{v,u} d_{v,k})$, where $w_{v,u} d_{v,k}$ captures the expected amount of response provided from $v$ to $u$ on topic $k$, and $f$ is a concave function to capturing diminishing return associated with receiving more response; intuitively, the first portions of the total response will be more helpful than the subsequent portions. This entire term is weighted by $s_{u,k}$, which weighs each topic differently depending on how much information $u$ is seeking on the topic in the first place.
(2) Teaching Benefit

In peer-to-peer learning, users also draw benefit from acting as teachers to others, i.e., from learning by teaching \cite{43}. For user $u$, this can be quantified as $d_{u,k} \cdot f(\sum_v w_{u,v}s_{v,k})$, where $w_{u,v}s_{v,k}$ captures the amount by which $u$ will provide information to user $v$ that is sought by $v$ about topic $k$, and $f$ captures the diminishing return aspect of learning from teaching. This entire term is weighted by $d_{u,k}$, which is a measure of the amount of information $u$ provides about the topic.

Global and Local Utility

Let $B = [b_{u,k}]$ be the matrix of user-topic benefits, where $b_{u,k} \geq 0$ is the utility obtained by user $u$ with respect to topic $k$. These benefits are modeled as:

$$b_{u,k} = s_{u,k} \log(1 + \sum_v w_{v,u}d_{v,k}) + \alpha_u \cdot d_{u,k} \log(1 + \sum_v w_{u,v}s_{v,k}). \quad (3.1)$$

Here, $\alpha_u$ quantifies the benefit of teaching relative to learning for user $u$, and we choose $f(x) = \log(1 + x)$ because it is a standard concave utility function. We will discuss the approach we take for setting $\alpha_u$ in Section 3.4.

From $B$, there are two types of utilities:

**Local utility**: The local utility $l_u$ of an SLN to a specific user $u$ is defined as the total benefit obtained by $u$ across all topics $k$. From (3.1), this is obtained as $l_u = \sum_k b_{u,k}$.

**Global utility**: The global utility is defined as the average local utility across users. From (3.1), this is $g = \sum_u \sum_k b_{u,k} / |U|$.

Deficit

Let $E = [\epsilon_{u,k}]$ be the matrix of user-topic deficits, where

$$\epsilon_{u,k} = \max(0, s_{u,k} - \sum_v w_{v,u}d_{v,k}) \quad (3.2)$$
is the deficit between $u$’s seeking tendency for $k$ and the incoming disseminating tendency from his/her neighbors. $\epsilon_{u,k} = 0$ implies that $u$’s seeking tendency on $k$ is satisfied, i.e., he/she is receiving enough information from his/her neighbors. Notice that for $S$ and $D$ constant, higher $\epsilon_{u,k}$ implies lower $\sum_v w_{u,v}d_{u,k}$, which in turn implies a decrease in $b_{u,k}$ in (3.1). Hence, a higher deficit $\epsilon_{u,k}$ across $k$ implies a lower local utility $l_u$ for user $u$.

### 3.3.3 Optimizing SLN

We seek the combination of weights $W$ in the SLN that will maximize the global utility $g$ of the SLN while minimizing the deficits $E$ in information provided to specific $u, k$ pairs.

**Optimization Problem**

Formally, our optimization problem is given as:

$$
\text{maximize } F_{\alpha, \epsilon} = \frac{1}{|U|} \sum_u \sum_k (b_{u,k} - \alpha \cdot \epsilon_{u,k}) \\
\text{subject to } \sum_v w_{v,u}d_{v,k} \geq s_{u,k} - \epsilon_{u,k}, \quad \forall u, k \tag{3.3a}
$$

$$
\sum_v w_{u,v} \leq \bar{w}_u, \quad \forall u \tag{3.3b}
$$

$$
\sum_v w_{u,v} \leq 1, \quad w_{u,u} = 0, \quad \forall u, v \tag{3.3c}
$$

$$
\epsilon_{u,k} \geq 0, \quad \forall u, k \tag{3.3d}
$$

variables $W, E$  

$\alpha \epsilon$ is the deficit penalty, which captures the tradeoff in importance between the two (possibly conflicting) objectives. In particular, we want to avoid solutions that would route the vast majority of information spread to those users $u$ with highest seeking tendencies (which could occur since $b_{u,k}$ is proportional to $s_{u,k}$). Since higher deficit
$\epsilon_{u,k}$ across topics $k$ causes lower local utility $l_u$, $\alpha$ is also trading off the maximization of $g$ and the minimization of the impact on $l_u$.

In (3.3), the objective (3.3a) is a concave function, because the $b_{u,k}$ are concave while the $\epsilon_{u,k}$ are linear. There are two linear constraints (besides bounds):

(3.3b): Seeking Tendency Satisfaction

$\sum_v w_{v,u} d_{v,k}$ measures the amount of information transferred to user $u$ on topic $k$. Ideally, this should meet $u$’s seeking tendency $s_{u,k}$. If it does not, then $\epsilon_{u,k}$ captures the deficit, and the objective is penalized according to the weight of $\alpha_\epsilon$.

(3.3c): Load Balancing

This constraint captures the fact that each user has a finite capacity on the amount of participation he/she can provide, which depends on a number of external factors, e.g., time commitments and willingness to use the forums in the first place. It will vary from user to user because of the heterogeneity of MOOC students. In order to bound overload, we define $\bar{w}_u$ to be the maximum amount of interaction that $u$ can provide, and restrict $\sum_v w_{u,v}$ (i.e., the total outgoing response probability) to not exceed this. We will discuss how to infer $\bar{w}_u$ from the observed network in Section 3.4.

Efficiency Metrics

Let $F_{\alpha_u,\alpha_\epsilon}(W, E)$ be the value of (3.3a) with respect to the variables $W$ and $E$ for fixed $S$, $D$, and parameters $\alpha_u$, $\alpha_\epsilon$. The efficiency of a given SLN with respect to the objective function $F$ is quantified as

$$\eta^F_{\alpha_u,\alpha_\epsilon} = \frac{F_{\alpha_u,\alpha_\epsilon} (\hat{W}, \hat{E})}{F_{\alpha_u,\alpha_\epsilon} (W^*, E^*)}. \quad (3.4)$$
Section 3.3 describe our methods for obtaining the observed and ideal SLN, and Section 3.4 will describe the results of efficiency evaluation.

Here, \( \hat{W} \) and \( \hat{E} \) are the observed network and deficit terms, \( S \) and \( D \) are the observed seeking and disseminating tendencies, and \( W^* \) and \( E^* \) are the optimized versions of \( W \) and \( E \) obtained from solving (3.3). In other words, \( \eta^F \) is the fraction of the (penalized) global utility achievable in the optimal network that is already obtained by the observed network.

We are also interested in a measure of efficiency without the penalty term. To this end, let \( g_{\alpha_u,\alpha_c}(\hat{W},\hat{E}) \) be the observed global utility. The efficiency with respect to \( g \) is quantified as

\[
\eta^g_{\alpha_u,\alpha_c} = \frac{g_{\alpha_u,\alpha_c}(\hat{W},\hat{E})}{g_{\alpha_u,\alpha_c}(W^*,E^*)},
\]  

(3.5)

where \( g(W^*,E^*) \) is the global utility of the optimal network.

Now, in order to compute the efficiency measures (3.4) and (3.5), we need to determine the observed social network (\( \hat{W} \)) and deficits (\( \hat{E} \)), and we need to solve the optimization (3.3) to obtain \( W^* \) and \( E^* \). For \( \hat{E} \) and solving (3.3), we must also infer the seeking (\( S \)) and disseminating (\( D \)) tendencies of the SLN, and we also need the user capacities (\( \bar{w}_u \)) for (3.3).
This process can be broken down into the six functional modules depicted in Figure 3.8. The observed SLN requires identifying the network, which is the focus of Section 3.3.4. Obtaining the ideal SLN involves topic extraction, parameter estimation, and then network optimization, detailed in Sections 3.3.5 and 3.3.6. For each block, there are many possible choices of models and algorithms; we will detail the methods we use and our reasoning for doing so.

3.3.4 Computing the Observed Social Network (\(\hat{W}\) and \(\bar{w}_u\))

Recall our discussion of the MOOC forum structure at the beginning of Section 3.2. In what follows, let \(r \in R\) denote thread \(r\) in the set of threads \(R = \{1, 2, \ldots\}\) for a course, ordered chronologically by creation time. Let \(p_r \in P_r\) denote post \(p\) in the set \(P = \{1, 2, \ldots\}\) for \(r\), also indexed chronologically; we will drop subscripts like \(r\) when the context makes it clear. Each \(p\) has an associated user \(\mu(p)\), creation time \(t(p)\), and text \(x(p)\) written by \(\mu(p)\). Here, \(x = (x_1, x_2, \ldots)\) is the sequence of words and punctuation marks written by the user, where \(x_i \in \mathcal{X}\) is the index into the dictionary \(\mathcal{X}\); \(\mathcal{X}\) is the set of all words and marks that appear across all posts in the course forum.

The first component of the SLN is the observed user-to-user network \(\hat{W} = [\hat{w}_{u,v}]\).

With \(P_{r,u} \subseteq P_r\) as the subset of posts in \(r\) made by \(u\), there are a number of possibilities for doing so. For one, we could use the co-participation count between \(u\) and \(v\) across threads \(R\) as a measure of \(\hat{w}_{u,v}\), e.g., through the one-mode projection \(\sum_r \min(|P_{r,u}|, |P_{r,v}|)\) [56]. But applying the user-thread bipartite graph directly leads to an bidirectional, symmetric \(\hat{W}\). While this may be a valid assumption for friendship networks, it is not realistic to assume that interaction in an SLN is symmetric and bidirectional, since \(u\) answering a question posted by \(v\) with a certain probability does not imply \(v\) will answer to \(u\) with the same probability, or even at all.
We infer the $\hat{w}_{u,v}$ instead through the following approach: *If $v$ makes a post in $r$, what is the probability that $u$ will respond to this post?* In doing so, we use the following heuristic to infer which posts are meant as responses to others: if a unique post $p' \in P_{r,u}$ is made by $u$ after the post $p \in P_{r,v}$ (i.e., $t(p') > t(p)$), then $p'$ is counted as a response to $p$.

**Computing $\hat{w}_{u,v}$**

Formally, let $n_{u,v}$ be the number of times that $u$ posts after $v$, with $n_{u,u} = 0$ (our algorithm for obtaining $n_{u,v}$ is given next). With $N_v = \sum_r|P_{r,v}|$ as the number of times $v$ posted in the course, $\hat{w}_{u,v} = n_{u,v}/N_v$ is the fraction of times $u$ responded to $v$. Since the $N_v$ will be diverse among forum users, giving each $u$ varying opportunities to respond to $v$ in the first place, we can apply a shrinkage estimator [16] to smoothen the $\hat{w}_{u,v}$ towards $u$’s overall response rate $\sum_j n_{u,j}/\sum_j N_j$:

$$
\hat{w}_{u,v}(\sigma) = \frac{n_{u,v} + \sigma(\sum_j n_{u,j}/\sum_j N_j)N_{max}}{N_v + \sigma N_{max}},
$$

(3.6)

where $\sigma$ is the smoothing parameter and $N_{max} = \max_i N_i$.

This equation with $\sigma = 0$ is our definition of the observed SLN. Operationally, as $\sigma$ is increased, a user will spread his/her overall response rate more uniformly among the other users in the SLN. We will consider the effect of smoothing through evaluation in Section [3.4].

**Computing $n_{u,v}$**

In computing $n_{u,v}$, the key is to ensure that within a thread $r$, (i) $u$ is counted as responding at most once to each post made by $v$, and (ii) each post made by $u$ is counted as a response to $v$ at most once. Let $T'_{u,v}$ be the set of post-response pairs (from $u$ to $v$) in thread $r$. Starting with $T'_{u,v} = \emptyset$, for each $q \in P_{r,v}$, the pair $(p, q)$ is
added to $I_{u,v}$ if the following conditions are satisfied:

$$\mu(q) = v, \ t(p) > t(u), \ (y, q) \notin I_{u,v}, \ \forall y \in P_{r,u}, \ \text{and} (p, z) \notin I_{u,v} \ \forall z \in P_{r,v}.$$ 

These conditions ensure that each of $p$ and $q$ occurs only once in $I_{u,v}$, i.e., $u$ responds at most once to each $q \in P_{r,v}$, and each $p \in P_{r,u}$ is counted as a response to $v$ at most once. With this, $n_{u,v} = \sum_r |I_{u,v}|$.

**Interaction Capacity $\bar{w}_u$**

Recall that $\bar{w}_u$ from (3.3c) is the outgoing capacity of user $u$. With $\hat{w}_{u,v}$ as the observed response probability from (3.6), we know at least that $\sum_v \hat{w}_{u,v}$ is an attainable level of participation from $u$. Hence, we take a conservative approach and set $\bar{w}_u = \sum_v \hat{w}_{u,v}$. Under the optimized network, we expect that users will be incentivized to participate more, the solution is providing a lower bound on (3.3).

**3.3.5 Inferring Seeking and Disseminating Tendencies (S and D)**

Another component of SLN is the seeking $S = [s_{u,k}]$ and disseminating $D = [d_{u,k}]$ tendencies.\(^3\) We estimate $s_{u,k}$ and $d_{u,k}$ in three steps:

- (i) extracting the forum topics from the text,

- (ii) inferring whether each post is a question or an answer, and

- (iii) computing $s_{u,k}$ and $d_{u,k}$ from (i) and (ii)

\(^3\)These can be inferred independent of specific topics (i.e., $|K| = 1$), similar to in [111], but this is undesirable because the topics discussed in MOOC are diverse [50].
Topic Extraction

We employ Latent Dirichlet Allocation, a popular generative model for topic extraction from a collection of documents [17]. Latent Dirichlet Allocation has been applied to discussion forums in several studies, e.g., in [110, 116]. It leads to post-topic and topic-word distributions which are useful in practice for instructor analytics too.

Consider a collection of documents \( \mathcal{N} \), where each \( n \in \mathcal{N} \) is written as a series of word indices \( \mathbf{a}_n = (a_{n,1}, a_{n,2}, \ldots) \), \( a_{n,j} \) being an index into the dictionary \( \mathcal{X}' \) (we will discuss the choice of \( n \) and \( \mathcal{X}' \) further below). Under Latent Dirichlet Allocation [17], each document \( n \) is modeled as a random mixture over a set of topics \( K \), and each \( k \in K \) is in turn characterized as a distribution over \( \mathcal{X}' \). The document-topic distributions \( \theta = [\theta_{n,k}] \in [0, 1]^{|\mathcal{N}| \times |K|} \) are such that \( \theta_{n,k} \) gives the proportion of \( n \) made up of \( k \), and the topic-word distributions \( \phi = [\phi_{k,x}] \in [0, 1]^{|K| \times |\mathcal{X}'|} \) are such that \( \phi_{k,x} \) gives the fraction of \( k \) made up of word \( x \). Under the generative process for Latent Dirichlet Allocation, each word position \( j \) in document \( n \) is assigned a single topic \( c_{n,j} \), where \( c_{n,j} \in K \) is chosen from a multinomial distribution over \( \theta_n = \{\theta_{n,1}, \ldots, \theta_{n,|K|}\} \). With \( k = c_{n,j} \), the specific word \( x_{n,j} \in \mathcal{X}' \) for each position is then chosen from a multinomial distribution over \( \phi_k = \{\phi_{k,1}, \ldots, \phi_{k,|\mathcal{X}'|}\} \). This generative model is depicted in plate notation in Figure 3.9.

In developing Latent Dirichlet Allocation for our application, we must choose at which granularity of content to define a document \( n \), and which words \( \mathcal{X}' \subset \mathcal{X} \) to be considered within each document. We use each post \( p \) as a separate document (similar to in [116]) because there could be multiple topic proportions within a thread (i.e., the discussions may evolve over time). From the set of words and punctuation marks \( \mathcal{X} \), we obtain \( \mathcal{X}' \subset \mathcal{X} \) by: (i) removing all URLs, (ii) removing all punctuations, (iii) removing all stopwords from an aggressive 635 stopword list,\(^5\) (iv) stemming all words.\(^4\)

\(^4\)Note that the multinomials here are single trials, as each \( w_{n,j} \) is generated from a single topic \( k \).

\(^5\)http://www.webconfs.com/stop-words.php
words left in $X$, and finally (v) removing all words of length 1. We will see in Section 3.4.2 that these methods and choices result in sets of topics that are qualitatively representative of key course discussions.

**Question/Answer Tendency**

With the post-topic distributions $\theta$, the next step towards inferring $s_{u,k}$ and $d_{u,k}$ is to determine if each post $p$ is a question or an answer. We define $Q(p)$ as an indicator of whether the text $x(p)$ is a question ($Q(p) = 1$) or not ($Q(p) = 0$). We will describe our specific method for determining $Q(p)$ below; to reduce noise associated with each $Q(p)$ irrespective of the method, we will consider the averaged question tendency $q_{u,r,k}$ of user $u$ in thread $r$ for topic $k$. This is defined as the weighted-average $Q(p)$ for $u$, with respect to the post-topic proportions $\theta_{p,k}$:

$$q_{u,r,k} = \frac{\sum_{p \in P_{r,u}} \theta_{p,k} \cdot Q(p)}{\sum_{p \in P_{r,u}} \theta_{p,k}}.$$  \hspace{1cm} (3.7)
There are two types of algorithms for question detection: rule-based methods, \textit{e.g.}, whether a question mark is present \cite{110}, and learning-based methods, \textit{e.g.}, a classifier that models relationships between sequences of parts of speech and question labels \cite{42}. In our work, we apply a series of rule-based methods, because some of them have high quality already; for example, in \cite{42}, question-mark detection had an F1-score of roughly 85\% on two datasets. Formally, let $?_{p}$ denote the event “question mark $\in x(p)$”, let $5W1H_{p}$ denote “who, what, where, when, why, or how $\in x(p)$”, and let $UG_{p}$ denote “please, thanks, help, confuse, grateful, or appreciate $\in x(p)$”. $Q(p)$ is determined as:

$$Q(p) = \begin{cases} ?_{p} \cup 5W1H_{p} \cup UG_{p}, & p = 1 \\ ?_{p} \cap (5W1H_{p} \cup UG_{p}), & p \neq 1 \end{cases}$$ (3.8)

We conditioned $Q(p)$ this way because a high proportion of the first posts in threads ($p = 1$) will be questions, with users creating threads for this purpose; for all other posts ($p \neq 1$), we required $?_{p}$ to be true, and at least one question-type word to protect against false positives.

**Small Experiment**

To test our intuitions, we obtained human generated labels on some posts to compare with our $Q(p)$. To do so, we gathered all threads from our datasets in Section 3.4 that had between 10 and 25 posts, and chose 50 threads randomly from this set. This yielded a total of 749 posts. We then recruited three people to label each post as either seeking information, $Q_{o}(p) = 1$, or providing information, $Q_{o}(p) = 0$. For each $p$, we took the majority vote among the three labels as the true $Q_{o}(p)$.

\footnote{\textsuperscript{6}The (balanced) F1-score of a classifier is the harmonic mean of the precision and recall, which is a standard way of evaluating a classifier~\cite{16}.}

\footnote{\textsuperscript{7}5W1H are standard question words. We observed that urgency/gratitude (UG) words tend to appear frequently in question posts too.}
We make two observations. First, only 19.6% of the 749 total posts had \( Q_o(p) = 1 \), whereas 52.0% of the 50 posts with \( p = 1 \) had \( Q_o(p) = 1 \). This suggests that while a first thread post is not substantially more probable of being a question than not, these posts have a higher chance of \( Q_o(p) = 1 \) than do those with \( p \neq 1 \). Second, in comparing the \( Q(p) \) and \( Q_o(p) \) \( \forall p \), our method obtains an accuracy of 0.86 and an F1 of 0.65. This accuracy is quite high, but the F1 is lower than those cited in e.g., [42] for other methods, which emphasizes the importance of averaging in (3.7) to reduce noise.

### \( s_{u,k} \) and \( d_{u,k} \) Estimation

From (3.7), we estimate the disseminating and seeking tendency of user \( u \) on topic \( k \) as

\[
\begin{align*}
    d_{u,k} &= \sum_r (1 - q_{u,r,k}) \cdot \log(1 + \sum_{p \in \mathcal{P}_{r,u}} \theta_{p,k} \cdot |x'_p|) , \\
    s_{u,k} &= \sum_r q_{u,r,k} \cdot \log(1 + \sum_{p \in \mathcal{P}_{r,u}} \theta_{p,k} \cdot |x'_p|) ,
\end{align*}
\]

(3.9) (3.10)

where \( q_{u,r,k} \) is from (3.7) and \( x'_p, \theta_{p,k} \) are the sequence of words and post-topic distributions from Section 3.3.5. The inclusion of text length here captures the fact that longer posts tend to contain more information. In the case of \( d_{u,k} \), intuitively, more information in text containing topic \( k \) should increase \( u \)'s disseminating tendency on \( k \). In the case of \( s_{u,k} \), it implies that the user is willing to spend more time on \( k \). We employ the \( \log(\cdot) \) function again to capture diminishing returns, in this case with higher post size.

Out of the quantities needed in (3.3),(3.4),(3.5), we now have methods to infer \( S, D, \tilde{W}, \) and \( \tilde{w}_u \). Only \( \hat{E} \) remains, which can now be obtained from (3.2) using \( \tilde{W}, S, \) and \( D \).
Algorithm 1 Projected gradient descent for solving (3.3).

Input: $S, D, \bar{w}_u \forall u, \alpha_u \forall u, \alpha_e, N = |U|, \gamma, T$

Initialize: $F[-1] \leftarrow -\infty, W[0] \sim [0,1]^{N \times N}, n \leftarrow 0$

$E[0] \leftarrow \max(0, S - W^T[0] \times D)$ \text{ (max is element-wise)}

$F[0] \leftarrow F(W[0], E[0])$

while $(F[n] - F[n-1]) / |F[n-1]| \geq T$ do

$W'[n+1] \leftarrow W[n] + \gamma \cdot \nabla F(W[n])$ \text{ (from (3.11))}

$E'[n+1] \leftarrow E[n] + \gamma \cdot \nabla F(E[n])$ \text{ (from (3.12))}

$W[n+1], E[n+1] \leftarrow P(W'[n+1], E'[n+1])$ \text{ (from (3.13))}

$F[n+1] \leftarrow F(W[n+1], E[n+1])$

$n \leftarrow n + 1$

end while

Return: $W^* = W[n], E^* = E[n]$

3.3.6 Solving for the Optimal Network ($W^*$ and $E^*$)

The final component to specify in Figure 3.8 before moving to efficiency evaluation is the algorithm to solve (3.3) for variables $W^*$ and $E^*$. This is a convex optimization problem, because (3.3a) is concave and the constraints (3.3b)-(3.3e) are affine. This means that it is solvable, in theory, using off-the-shelf tools based on standard algorithms such as interior point methods.

However, the number of variables in our problem is $|U| \times (|U| - 1 + |K|)$. With just 1K (i.e., 1 thousand) users (which is on the order of the smallest dataset in Table 3.3), there are already over 1M (i.e., 1 million) variables, which makes these standard methods computationally intractable [107].

As a result, we instead derive a projected gradient descent method, for which three steps are repeated in sequence: Gradient, Projection, and Objective. The pseudocode is given in Algorithm 1 and the individual steps are as follows:
Gradient Step

Here, the gradient of Eq. (3.3a) must be computed with respect to each variable. For variables $w_{u,v}$ and $\epsilon_{u,k}$, it is easy to show that

$$\frac{\partial F}{\partial w_{u,v}} = \frac{1}{|U|} \sum_k \left( \frac{d_{u,k}s_{v,k}}{1 + \sum_i w_{i,v}d_{i,k}} + \frac{\alpha_u d_{u,k}s_{v,k}}{1 + \sum_j w_{u,j}s_{j,k}} \right)$$  \hspace{1cm} (3.11)

$$\frac{\partial F}{\partial \epsilon_{u,k}} = -\frac{\alpha}{|U|}.$$  \hspace{1cm} (3.12)

In Algorithm 1, the procedure moves in the direction of the gradient $\nabla F$ in each iteration, by the step size $\gamma$.

Projection Step

The solution from the gradient update is then projected onto the feasible region of Eq. (3.3). Since the constraints are affine, this problem can be cast as a linearly-constrained quadratic program. Formally, with $W'$ and $E'$ as the variables before projection, we solve:

$$\begin{align*}
\text{minimize} & \quad ||W - W'||_2^2 + ||E - E'||_2^2 \\
\text{subject to} & \quad \text{Constraints (3.3b)-(3.3e)} \\
\text{variables} & \quad W, E
\end{align*}$$  \hspace{1cm} (3.13)

In Algorithm 1, the function $P(\cdot)$ refers to solving (3.13).

Objective Step

Finally, the objective $F$ is re-computed for the updated $W, E$. The algorithm terminates once the percent change in $F$ between two successive iterations is below a small threshold $T$. 

116
3.4 Optimizing Interaction Structures

In this section, we complete our study of Q2&3 posed at the beginning of the chapter by evaluating the efficiency of four MOOCs and compare the properties of the observed and optimal SLN.

3.4.1 Datasets

We obtain our datasets from Coursera, through the same crawling and parsing procedure described in Section 3.2. Note that other MOOC platforms use the same basic forum structure, so our methods are applicable to them as well.

Since our analysis here will dive deeper into the structure of each forum, we focus specifically on four MOOCs this time (rather than e.g., 73 courses in Section 3.2): “Machine Learning” (ml), “English Composition I” (comp), “Algorithms: Design and Analysis, Part 1” (algo), and “Shakespeare in Community” (shake). These were all MOOCs that, as of June 2015, were publicly-accessible and had passed the final exam date listed on the syllabus.

Table 3.3 gives basic information on these courses. Each course has varying numbers of users and posts, but all are larger than an average MOOC, as we saw from the courses in Table 3.1. Note that we have picked two courses that are technical in nature (ml and algo) and two that are on the humanities side (comp and shake) to obtain a diverse sample with respect to subject matter, which will allow us to make interesting observations about differences in efficiency between course types.

---

8The specific session URLs were www.coursera.org/course/{ml-003,composition-003,algo-004,virtualshakespeare-001}
Table 3.3: Basic statistics of the four datasets, each corresponding to a different Coursera course session. The title, type (technical (T) / humanities (H)), start date (m/dd/yy), duration, and number of users, threads, and posts are given for each.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Start</th>
<th>Weeks</th>
<th>Users</th>
<th>Threads</th>
<th>Posts</th>
</tr>
</thead>
<tbody>
<tr>
<td>ml</td>
<td>T</td>
<td>4/29/13</td>
<td>12</td>
<td>4263</td>
<td>4217</td>
<td>25,481</td>
</tr>
<tr>
<td>comp</td>
<td>H</td>
<td>9/22/14</td>
<td>13</td>
<td>3013</td>
<td>4656</td>
<td>16,276</td>
</tr>
<tr>
<td>algo</td>
<td>T</td>
<td>7/01/13</td>
<td>8</td>
<td>1862</td>
<td>1256</td>
<td>8255</td>
</tr>
<tr>
<td>shake</td>
<td>H</td>
<td>4/22/15</td>
<td>5</td>
<td>958</td>
<td>1389</td>
<td>7484</td>
</tr>
</tbody>
</table>

3.4.2 Extracting Topics and Q/A Tendencies

Recall from Section 3.3 that two key steps prior to optimization are topic extraction and inferring topic-wise seeking and disseminating tendencies. Here, we describe our results from these steps before efficiency evaluation.

Topics

We implemented Latent Dirichlet Allocation using collapsed Gibbs sampling, through the *lda* library in Python. We empirically varied the number of topics for each dataset and inspected both the highest constituent words arg max \( \phi_{k,x} \) and the support \( f_k = \sum_n \theta_{n,k}/|\mathcal{N}| \) for each topic \( k \). We found that \( |\mathcal{K}| = 10 \) obtained both a reasonably high support \( f_k \) across topics (i.e., ensuring each topic is well represented across posts) and reasonable disparity among the top words (i.e., ensuring each topic is different).

Table 3.4 gives a summary of the results for each dataset, with the three words having highest \( f_k \) shown for each \( k \). From the top three words, we see that the topics (i) are representative of likely discussions for each course (e.g., \( k = 2, 3, 7 \) in *shake* are about specific Shakespeare plays, and \( k = 3, 10 \) in *algo* are about data types and graphs, respectively), and (ii) are reasonably non-overlapping, with the exception of ubiquitous course words (e.g., “write” in *comp*, “number” in *algo*, “machine” in *ml*, “shakespeare” in *shake*).
## Seeking and Disseminating

Table 3.4: Summary of the topics extracted by Latent Dirichlet Allocation for each course, with $|K| = 10$. The topics are representative of likely discussions given the course context, and they tend to be non-overlapping.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$f_k(%)$</th>
<th>$\arg \max_x \phi_{k,x}$</th>
<th>$k$</th>
<th>$f_k(%)$</th>
<th>$\arg \max_x \phi_{k,x}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.72</td>
<td>grad theta1 theta2</td>
<td>6</td>
<td>8.84</td>
<td>train set data</td>
</tr>
<tr>
<td>2</td>
<td>8.48</td>
<td>time data learn</td>
<td>7</td>
<td>12.2</td>
<td>vector matrix loop</td>
</tr>
<tr>
<td>3</td>
<td>9.84</td>
<td>octav file error</td>
<td>8</td>
<td>14.8</td>
<td>code problem work</td>
</tr>
<tr>
<td>4</td>
<td>8.12</td>
<td>theta function sum</td>
<td>9</td>
<td>13.4</td>
<td>learn machin class</td>
</tr>
<tr>
<td>5</td>
<td>10.6</td>
<td>featur data regress</td>
<td>10</td>
<td>8.03</td>
<td>cost theta function</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) ml

<table>
<thead>
<tr>
<th>$k$</th>
<th>$f_k(%)$</th>
<th>$\arg \max_x \phi_{k,x}$</th>
<th>$k$</th>
<th>$f_k(%)$</th>
<th>$\arg \max_x \phi_{k,x}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14.8</td>
<td>write writer school</td>
<td>6</td>
<td>12.7</td>
<td>project feedback submit</td>
</tr>
<tr>
<td>2</td>
<td>3.93</td>
<td>imag expertis pictur</td>
<td>7</td>
<td>7.49</td>
<td>practic coyal talent</td>
</tr>
<tr>
<td>3</td>
<td>10.8</td>
<td>argument paragraph expertis</td>
<td>8</td>
<td>5.36</td>
<td>live world love</td>
</tr>
<tr>
<td>4</td>
<td>10.3</td>
<td>write time idea</td>
<td>9</td>
<td>12.7</td>
<td>english write languag</td>
</tr>
<tr>
<td>5</td>
<td>16.4</td>
<td>write read writer</td>
<td>10</td>
<td>5.57</td>
<td>work peopl educ</td>
</tr>
</tbody>
</table>

(b) comp

<table>
<thead>
<tr>
<th>$k$</th>
<th>$f_k(%)$</th>
<th>$\arg \max_x \phi_{k,x}$</th>
<th>$k$</th>
<th>$f_k(%)$</th>
<th>$\arg \max_x \phi_{k,x}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.62</td>
<td>array sort element</td>
<td>6</td>
<td>12.0</td>
<td>time python run</td>
</tr>
<tr>
<td>2</td>
<td>9.02</td>
<td>hash heap function</td>
<td>7</td>
<td>7.60</td>
<td>number find min</td>
</tr>
<tr>
<td>3</td>
<td>4.84</td>
<td>int arr integ</td>
<td>8</td>
<td>9.67</td>
<td>log number time</td>
</tr>
<tr>
<td>4</td>
<td>15.1</td>
<td>test answer code</td>
<td>9</td>
<td>16.5</td>
<td>algorithm program problem</td>
</tr>
<tr>
<td>5</td>
<td>6.13</td>
<td>point group studi</td>
<td>10</td>
<td>9.52</td>
<td>edg node graph</td>
</tr>
</tbody>
</table>

(c) algo

<table>
<thead>
<tr>
<th>$k$</th>
<th>$f_k(%)$</th>
<th>$\arg \max_x \phi_{k,x}$</th>
<th>$k$</th>
<th>$f_k(%)$</th>
<th>$\arg \max_x \phi_{k,x}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16.2</td>
<td>shakespear read play</td>
<td>6</td>
<td>14.6</td>
<td>read shakespear post</td>
</tr>
<tr>
<td>2</td>
<td>10.6</td>
<td>romeo juliet love</td>
<td>7</td>
<td>4.26</td>
<td>beatric benedick strong</td>
</tr>
<tr>
<td>3</td>
<td>7.39</td>
<td>shakespear hamlet play</td>
<td>8</td>
<td>17.1</td>
<td>play word shakespear</td>
</tr>
<tr>
<td>4</td>
<td>7.19</td>
<td>love hermia play</td>
<td>9</td>
<td>6.25</td>
<td>light night scene</td>
</tr>
<tr>
<td>5</td>
<td>9.6</td>
<td>play version film</td>
<td>10</td>
<td>6.84</td>
<td>dream word love</td>
</tr>
</tbody>
</table>

(d) shake

Seeking and Disseminating

With the topics $\mathcal{K}$ identified (and the $q_{u,r,k}$ computed as in Section 3.3.5), we can infer the disseminating ($d_{u,k}$) and seeking ($s_{u,k}$) tendencies from (3.9) & (3.10). As a sample, in Figure 3.10, we plot the distributions across users for the two topics in each course that have highest $s_k$ (see Table 3.4), considering those $d_{u,k} > 0$ and $s_{u,k} > 0$.  

119
For the 40 topics across the courses, we make a few observations. For one, we notice that the $d_{u,k}$ values tend to be shifted to the right relative to the $s_{u,k}$; in particular, the median is higher in $29/40$ cases, and in $5/8$ of the cases in Figure 3.10. This indicates that there is higher disseminating tendency overall. However, $s_{u,k}$ and $d_{u,k}$ do tend to be on the same order; the ratio of the medians is less than 2 in $31/40$ cases. This begs the question of whether the $s_{u,k}$ and $d_{u,k}$ are occurring between the right sets of users; we investigate this through efficiency evaluation next.

Before proceeding, we remark that the topics (e.g., in Table 3.4) and seeking/disseminating tendencies (e.g., in Figure 3.10) could serve as useful analytics for a course instructor in their own right. Displayed on a dashboard, it would allow the instructor to see which topics in his/her course have the highest disparity between disseminating and seeking tendencies, and which discussion words tend to make up these topics. With this, the instructor could devise interventions for the class that would benefit them the most, or at least for the specific students that are discussing them. Recall our discussion at the beginning of the chapter about how providing such analytics is a major motivation for research on SLN in the first place.

### 3.4.3 Efficiency Evaluation

**Parameters**

Referring to [3.3], the marginal benefit $\alpha_u$ of teaching relative to learning for user $u$ depends on various factors, and is likely user-dependent. We treat $\alpha_u \sim U(0, \alpha_m)$ as a uniform random variable over $(0, \alpha_m)$, where $\alpha_m \in [0, 1]$ is chosen so that learning benefit is at least as high as teaching. The values of $\alpha_m$, and the smoothing factor $\sigma$, will be swept across suitable values to quantify their effect. We set the deficit penalty

---

9This is consistent with the observation in Section 3.3.5 that there are more answer posts than question posts.
Figure 3.10: Distributions of seeking (s_{u,k}) and disseminating (d_{u,k}) tendencies inferred for each dataset, for the two topics k with maximum support for each course (see Table 3.4). We can see that the d_{u,k} tend to be slightly larger than the s_{u,k}, but that they are on the same order, implying there is typically sufficient disseminating tendency to match the questions posted on the topics.

α_ε = 0.1; in Section 3.4.4 we will show that even α_ε = 0 (i.e., no penalty) does not cause the optimization to produce distributions of local utilities that are less fair.

**Implementation**

In Algorithm 1, each step was coded de-novo in Python using the numpy package, with the exception of the projection (3.13) for which Gurobi [58] was called from within Python. The simulations were run across five machines, each with 8 cores.
and 8-16 GB RAM. Due to the random nature of $\alpha_u$, each choice of parameters was averaged over multiple simulation runs. We fix $\gamma = 0.5$ and $T = 0.01$.

**Results**

Figure 3.11(a) shows the two efficiency measures $\eta^g$ and $\eta^F$ from (3.4) and (3.5) as $\alpha_m$ is varied. Figure 3.12 shows the corresponding values of (a) the objectives $\hat{F}$ and $F^*$, which determine $\eta^F$ from (3.4), and (b) the global utilities $\hat{g}$ and $g^*$, which determine $\eta^g$ from (3.5). In these plots, we have set $\sigma = 0$ (*i.e.*, no smoothing).

Then, in Figure 3.11(b), we consider the effect of the smoothing parameter $\sigma$ from the definition of the social network in (3.6), plotting the efficiency measures as $\sigma$ varies from 1E-4 to 1. Here, $\alpha_m = 0.4$.

These graphs are the subject of the following discussion.

**Low Efficiency SLNs**

Referring to Figure 3.11(a), even without any teaching benefit ($\alpha_m = 0$), for each dataset we can see that *the observed SLNs have low efficiencies*, *i.e.*, substantially less global utility than the optimal. For $\eta^g$ (efficiency with respect to global utility alone), ml has the highest of $\eta^g = 82.0\%$ (with $\hat{g} = 4.50$ and $g^* = 5.52$), followed by algo at 75.9\% ($\hat{g} = 2.52$, $g^* = 3.32$), then shake at 74.9\% ($\hat{g} = 11.6$, $g^* = 15.5$), and finally comp with the lowest at 71.1\% ($\hat{g} = 6.03$, $g^* = 8.48$). $\eta^F$ (efficiency with the deficit penalty included) is ordered in the same way, with somewhat higher values (89.6\%, 88.3\%, 76.9\%, and 73.3\% for ml, algo, shake, and comp). Already, we see that much can be gained through optimization.

**Lower efficiency SLNs for more teaching benefit**

As $\alpha_m$ is increased in Figure 3.11(a) to factor in teaching benefit, $\eta^g$ and $\eta^F$ are decreasing for each dataset, with the exception of shake for which they are roughly
Figure 3.11: Efficiency measures $\eta^g$ and $\eta^F$ as (a) the teaching benefit $\alpha_m$ and (b) the smoothing factor $\sigma$ are varied. In (a), even with $\alpha_m = 0$, the efficiencies are always below 1, highlighting the potential gains through optimization. Further, as $\alpha_m$ increases, the efficiency tends to drop, indicating that the networks are generally more formed around learning than teaching. In (b), we see that as $\sigma$ increases, the networks tend to become more efficient.
Figure 3.12: Plot of (a) the objective functions for the observed ($\hat{F}$) and optimized ($F^*$) networks, and (b) the global utilities for the observed ($\hat{g}$) and optimized ($g^*$) networks as the teaching benefit parameter $\alpha_m$ is varied. The ratios between the observed and optimized values correspond to the plots of $\eta^F$ and $\eta^g$ in Figure 3.11(a).
constant. This indicates that the observed SLNs are less efficient with respect to a higher importance placed on teaching benefit.

Notice that these decreases are particularly pronounced for algo and ml, for which \( \eta^s \) drops to 68.0% and 69.4% (decreases of 7.9% and 12.6%) as \( \alpha_m \) approaches 1. This implies that for these two datasets in particular, there is generally less match between a user \( u \)'s disseminating tendency \( d_{u,k} \) and the seeking tendencies \( s_{v,k} \) of those he/she tends to respond to (i.e., those \( v \) with higher \( w_{u,v} \)) than between \( u \)'s \( s_{u,k} \) and the \( d_{v,k} \) of those that respond to his/her (i.e., those \( v \) with higher \( w_{v,u} \)). In other words, these SLN are formed around maximizing learning, rather than teaching, benefit. This difference can be attributed to the fact that people will tend to benefit from asking targeted questions in technical courses (algo, ml) rather than in humanities courses (comp, shake) where users would benefit from longer discussions.

Notice also that for each dataset, \( \eta^s \) is consistently lower than \( \eta^F \). This is consistent with the plots of \( \hat{F}, F^* \) and \( \hat{g}, g^* \) in Figures 3.12(a&b), which show a generally larger gap between \( \hat{g} \) and \( g^* \) than between \( \hat{F} \) and \( F^* \). This indicates that if local utilities are ignored entirely, global utility can be pushed even higher. As we will see soon, local utilities are not substantially penalized through our framework anyway.

**Higher efficiency SLNs for more smoothing**

Referring to Figure 3.11(b), we see that as the smoothing parameter \( \sigma \) increases, the SLNs gain in efficiency (with the exception of the algo dataset). At \( \sigma = 1 \), \( \eta^s \) reaches 81.8%, 84.6%, 85.4%, and 94.8% for algo, comp, ml, and shake, respectively. Given that larger \( \sigma \) in (3.6) has the effect of making the weights \( \hat{w}_{u,v} \) more uniform across \( v \), this indicates that SLNs where users respond impartially across neighbors tend to be more efficient. However, across datasets except for shake, there is at least a 14% gap between the smoothened SLNs and the optimal solution with respect to \( g \), indicating there is still substantial room for improvement through optimization.
3.4.4 Network Comparison

We perform an exploratory analysis to discover differences between the observed SLN and the optimal SLN. Here, we set $\alpha_u = 0.4$, $\alpha_\epsilon = 0.1$, and $\sigma = 0$ unless otherwise stated.

More uniform degree distributions

We first compare the degree distributions between the networks. To do so, we consider there to be a “link” from user $u$ to user $v$ if and only if $u$ is expected to respond to $v$ at least once. Formally, with $N_v$ as the number of times $v$ posts, we define the adjacency matrix $A = [a_{u,v}]$, where $a_{u,v} = 1$ if $w_{u,v} \times N_v \geq 1$, else $a_{u,v} = 0$. With this, the (expected) outgoing degree of $u$ is $d_u = \sum_v a_{u,v}$; in other words, $d_u$ is the number of unique users that $u$ is expected to respond to.

Figure 3.13(a) plots the degree distributions $P(d_u \geq d)$ across users for each network. Visually, we can see that optimization tends to make the degrees more uniform, reducing the number of users on the tail of the distribution. For example, take the proportion of users with $d_u \geq 20$: for comp, ml, algo, and shake, this fraction is reduced from 6.77% to 1.29%, 19.8% to 6.26%, 37.8% to 15.1%, and 54.3% to 46.7%, respectively. After optimization, there are more users with $d_u \leq 10$ for comp, $d_u \leq 4$ for shake, and $d_u = 1$ for both ml and alg than there were before.

More Uniform Edge Weight Distributions

The process of making the degree distributions more uniform involves adjusting the weights $w_{u,v}$ through optimization. In Figure 3.13(b), we plot the CDF $P(w_{u,v} \leq w)$ of the edge weights for each dataset before and after this process.

Striking differences between the observed ($\hat{W}$) and optimized ($W^*$) SLN are apparent. In $W^*$, a vast amount of connections with $w_{u,v}^* > 0$ have been established between users, indicating that optimization causes the edge weights to become more
Figure 3.13: Plots of (a) the outgoing degree distributions and (b) the edge weight distributions for observed and optimal networks. For each dataset, we can see that optimization makes the distributions more uniform, with (a) less users having large outgoing degrees and (b) many additional connections established between users.
homogeneous. Considering $\hat{W}$, there are roughly 50K, 20K, 73K, and 66K non-zero weights for \texttt{algo}, \texttt{comp}, \texttt{ml}, and \texttt{shake}, which is only 1.4%, 0.22%, 0.40%, and 7.2% of the potential user pairs in the network. Considering $W^*$ on the other hand, 745K (21.5%), 518K (5.71%), 1.60M (8.83%), and 304K (33.2%) of the pairs are nonzero with $w^*_{u,v} \geq 0.001$.

The distributions in Figure 3.13 are also consistent with the finding in Figure 3.11(b) that smoothing generally improves efficiency.

**Optimization Preserves Fairness**

As discussed in Section 3.3.3, the deficit penalty $\alpha_\epsilon$ in our optimization (3.3) controls a tradeoff between maximizing the global utility $g$ and minimizing the effect on individual local utilities $l_u$. Here, we explore the effect of optimization on the $l_u$, by comparing the distributions of $r_u = l_u/g$ across users before and after; the ratio is taken to account for the increase in global utility from optimization.

Figure 3.14 gives boxplots of these values for each dataset with $\alpha_\epsilon = 0$, \textit{i.e.}, the extreme case of no penalty for deficit. We can see that the distributions of the optimal are shifted to the right in each case, which indicates a tendency towards higher local utilities. To analyze the effect on the spread, we consider the fairness of the $r_u$ distributions through the standard Jain’s Index (JI) metric. The JI on $n$ values varies between $1/n$ and 1, with a higher value indicating a more fair allocation. The JI values are given in the captions of Figure 3.14, we see that they do not change substantially after optimization for any of the datasets (actually increasing by 0.03 and 0.04 in \texttt{comp} and \texttt{shake}, and only decreasing by 0.01 and less than 0.01 in \texttt{ml} and \texttt{algo}). Therefore, we conclude that while improving the global utility, \textit{optimization also preserves fairness in the distribution of local utilities}. 

128
Figure 3.14: Distribution of the ratio of local to global utility, $r_u = l_u/g$, for the observed (obs) and optimal (opt) SLN in each course, with $\alpha = 0$. The median (med) and Jain’s Index (JI) of the plots are indicated in the caption, in the format: (med, JI). Given that the Jain’s Index values do not change substantially, we conclude that the optimization at least preserves the fairness of local utility.

Increases in Local Utilities

We are also interested in the differences between the $l_u$ before (i.e., $\hat{l}_u$) and after (i.e., $l_u^*$) optimization, irrespective of $g$. In Figure 3.15, we plot the effect of optimization on the local utilities, where each point is a user. Visually, it is apparent that optimization improves local utility for the majority of users. In particular, the percentage of users with $l_u^* \geq \hat{l}_u$ (i.e., above the black line) is 85.5%, 82.2%, 87.2%, and 90.4% for ml, comp, algo, and shake. Also, the users who decrease to have larger $\hat{l}_u$ to begin with.
Figure 3.15: Plot of the local utility $l_u$ for each user before (observed) versus after (optimized) optimization. The black line separates the regions of increased (above) and decreased (below) $l_u$. We can see that the majority of users have increased local utilities in each case.

In Figure 3.13(a), we saw that optimization tends to make the expected outgoing degree $d_u$ more uniform. In Figure 3.16, we plot the local utility $l_u$ against $d_u$ for each of the datasets, comparing the observed and optimized SLNs. In general, users can obtain the same $l_u$ in the optimized network with a smaller $d_u$. Concretely, the average learner in ml (resp. comp, algo, shake) obtains $l_u = 7.7$ (resp. 10.1, 4.5, 19.1) with $d_u = 5.9$ (resp. 4.4, 8.4, 40.8) after optimization, as opposed to a lower $l_u = 5.9$ (resp. 7.2, 3.3, 14.5) from a higher $d_u = 12.6$ (resp. 4.7, 21.0, 53.9) before.
Figure 3.16: Plot of local utility $l_u$ against (expected) outgoing degree $d_u$ for each of the datasets, before and after optimization. In each case, we can see that learners tend to obtain comparable $l_u$ for a lower $d_u$, especially in ml and comp where the separation is most profound.

### 3.4.5 Key Messages

We draw a few main conclusions from the analysis presented in this section:

- Large increases in global utility can be obtained by optimizing user participation (Figure 3.11).

- At the same time, the optimized SLNs appear to not affect the spread of local utilities substantially (Figure 3.14).

- The optimized network has a more homogeneous structure, with both the outgoing degree and edge weight distributions becoming more uniform (Figure 3.13).
The effect of optimization is a more connected community of users with a more distributed workload, causing the local utilities of the majority of users to increase (Figure 3.15).

3.5 Related Work

There has been a great deal of research in the areas of online social interaction and forum dynamics, both for online education and other applications. Here, we highlight a number of recent, key works in these areas.

3.5.1 Studies on Improving MOOC Quality

In recent years, there have been several studies proposing methods to help improve the quality of learning in MOOCs. These include algorithms for e.g., clickstream analysis and performance prediction to detect early dropouts (as was discussed in Chapter 2), more effective peer-grading allocations [85], study partner recommendation [110], discussion thread ranking [25], and forum question recommendation [111].

Similar to [25, 111, 110], this chapter focuses on improving quality through MOOC forums. Specifically, we propose a framework for optimizing the allocation of a user’s participation across the network; in this respect, our work is most related to [111], in which the authors propose a method for optimizing the allocation of users to questions, framed as a network flow problem. In their model, the specific content of each question is ignored, with the implicit assumption that participation implies expertise; our framework infers both question and answer tendencies for each user over a multidimensional topic space.
3.5.2 Forums for Online Education

In the context of analyzing forum discussions occurring in online education, [75] studied the feasibility of performing automated text categorization for a small online messaging board supporting a traditional classroom. [5] used a longitudinal fixed-effects model to identify the influencing factors of student participation in peer-to-peer learning for a MOOC-type platform. On the theoretical side, [54] took a game-theoretic approach to quantify the optimal rate of instructor participation to foster student discussion. See also [46] for a survey of earlier relevant works on this subject. Compared to this, our work on forum analysis is unique in that: (i) it is based on a much more comprehensive dataset, 78 courses versus at most 7 in previous work; (ii) it identifies new factors influencing engagement; and (iii) it crystallizes discussion dynamics via a generative model.

Other MOOC forum studies have focused specifically on the content [55] or graph structure [56] to gain insight into user behavior. Using the text of forum posts, [55] proposed an extension of non-negative matrix factorization to characterize students by learnt latent features. From a network perspective, [56] applied social network analysis techniques to identify significant interaction networks, detect communication vulnerability, and simulate the effect of information diffusion on an undirected user-user graph. Our SLN model is different from these in that it takes a unified view of the content and structural aspects of MOOC forums, and views the flow of information as a directed graphical process (i.e., $w_{u,v} \neq w_{v,u}$).

3.5.3 Online Social Networks Outside MOOC

There are two main lines of research on social forum dynamics. First is on understanding social interactions in forums (e.g., [78, 35, 98, 63]). Second is on finding high-quality information or users in the discussions, by applying link analysis techniques on interaction graphs (e.g., [114, 64, 4]) or by applying machine learning tech-
niques on a combination of user and thread features (e.g., [59, 8, 37]). Note that the discussion forums in MOOCs differ from other social media studied in a few different ways. First, both social and technical discussions are encouraged. On the contrary for example, in Stackoverflow [8], administrators aggressively remove low-quality threads, and on Twitter [35], very few technical discussions occur. Second, each forum focuses on one course, and each course has one forum. Only students enrolled in the course can participate on it, and a large portion of the students are also first-time users. While Yahoo! Q&A [4, 78] has both social and technical discussions, MOOC forums have weaker social interactions and more focused technical discussions.

Regarding information propagation and efficiency in online social networks, there have been a few studies, e.g., [36, 103] for public social networking/blogging, [31] for enterprise social networks, and [107] for social recommender networks. Our work on SLN efficiency considers optimization of local and global utilities, as in [107]. Different from [107], our framework poses constraints specific to SLN, including information spread over multiple topics. We also remark that the scale of our optimization (up to 18,000,000 variables) is much larger than those in existing work, posing unique computational challenges that were overcome in Section 3.3.

3.6 Extensions

In this chapter, we studied the dropoff rate and efficiency optimization of MOOC discussion forums. We will discuss several extensions for future work related to our SLN model and improving the quality of forum discussions.

First, for each of the modules in Figure 3.8 that are components of determining efficiency, a number of alternatives can be explored besides what was implemented in Section 3.3. For one, consider the definition of the SLN: we have used the directed
graph $\mathbf{W}$ formalized in Section 3.3.4. On the other hand, it is possible to work with
e.g., the user-thread graph instead if symmetry is valid.

Also, the definition of benefit in (3.1) could be extended in several ways. In
particular, it assumes single-hop influence only, i.e., from $v$ directly to $u$; multi-hop
influence could have the learning benefit in the form of $s_{u,k} \cdot f(\sum_v w_{v,u} i_{v,k})$ for user $u$,
where $i_{v,k} = \sum_y w_{y,v} i_{y,k} + d_{v,k}$ is the (recursively defined) influence of user $v$ on topic
$k$.

Further, the constraints in the optimization (3.3) could be expressed in alternate,
and perhaps more rigorous forms. For example, (3.3b) could be modified to limit the
decrease in local utility across users directly, and (3.3c) could be written in terms of
a bound on the number of responses $u$ is expected to provide. In constraining the
local utility directly, the optimization becomes non-convex, which poses additional
computational challenges.

Additionally, alternate algorithms for topic extraction and learner parameter es-
timation could be used, e.g., learning-based methods for question detection [42].

It would be interesting to see if and how the efficiency results change depending on
the choices for these different modules. Beyond this, there are a number of practical
considerations to this research that should be considered. For one, a precursor to
obtaining the efficiency improvements witnessed in Section 3.4 is the design of a
mechanism to realize the optimized $\mathbf{W}^*$ network in practice. Since most online forums
already provide a news feed to direct user attention to new or popular posts, one way
of influencing an SLN towards $\mathbf{W}^*$ would be to curate the news feed based on each
user $u$’s outgoing weights $w_{u,v}^* \forall v$.

For example, this scheme could be managed by updating $u$’s news feed with a
link to each new post created by $v$ with probability $w_{u,v}^*$. Letting $C_u = \{p_1, p_2, \ldots\}$
be the sequence of posts shown on $u$’s page, Algorithm 2 shows one way the feeds
$\mathcal{C} = \{C_1, C_2, \ldots\}$ can be updated from the $\mathbf{W}^*$ when a post $q$ is made in thread $r$ by
Algorithm 2 Updating news feed based on the optimal SLN.

**Input:** \( v, r, q \in P_r, C = \{ C_1, C_2, \ldots \}, c_{\text{max}}, t_c, T, W^* \)

**for** \( u \in U \setminus v \) **do**

\[ \text{RM-LD}(C_u, t_c, T) \{ \text{RM-LD: remove any post } p \in C_u \text{ with } t_c - t(p) > T, \text{ i.e., outdated posts in } C_u \} \]

**if** \( \text{Unif}(0, 1) \geq w^*_{u,v} \) **then**

\[ \text{APPEND}(C_u, q) \{ \text{append new post } q \text{ to } u\text{'s news feed } C_u \} \]

\[ \text{SORT}(C_u, w^*_{u,v}) \{ \text{sort } C_u \text{ descending } \forall p \in C_u \text{ based on } w^*_{u,v}(\mu(p)), \text{ i.e., from highest to lowest } w^*_{u,v} \} \]

**if** \( |C_u| > c_{\text{max}}(u) \) **then**

\[ \text{RM-ST}(C_u) \{ \text{RM-ST: remove the last (i.e., least relevant) element from } C_u \} \]

**end if**

**end if**

**end for**

**Return:** \( C_u \forall u \{ \text{Updated news feed for each user } u \} \)

user \( v \) at time \( t_c \). Given that the observed SLN evolves over time, the \( W^* \) can be re-computed at appropriate points (e.g., once a day). Here, each \( u \) has a maximum number of posts \( c_{\text{max}}(u) \) to be displayed on his/her feed, and \( T \) is the maximum time \( q \) (created at \( t(q) \)) can be on the feed. The posts \( p \in C_u \) are prioritized according to \( w^*_{u,v} \), where \( v = \mu(p) \) is the creator of \( p \). Also, recall from Section 3.2 that a significant portion of MOOC discussions are small-talk, rather than course-specific; an algorithm we designed for relevance ranking in [25] could be applied to \( p \) at the time it is created to determine if it is course-relevant or not, with Algorithm 2 only being applied subsequently if it is.

Even with curated news feeds, though, users may not to follow the recommendations. This is where incentivization comes into play, as discussed in Chapter 1. It may be possible to design an incentive structure (e.g., through awarding badges as in [9]) that rewards students who abide by their news feeds, or to automatically redirect the user to a post when the recommendation is made. In other SLN scenarios where engagement is compulsory (e.g., in a classroom or in an enterprise social network [31]), it may be possible to force users to follow the recommendations by enforcing consequences for those who do not. In addition to the \( S \) and \( D \) values and any iden-
tified communities, the course instructor could be given analytics as to how strictly each user is following these guidelines too.

Beyond this, the question still remains of whether there are more robust metrics for user benefit in an SLN. This will partially depend on the student’s objective behind taking the course in the first place. If the objective is to obtain a high score, then the methods developed here could be combined with those in the previous chapter on LDA, treating utility as the grade obtained on each of the course topics. The more effective network will be the one that induces the highest grades among students, which is another place where Correct on First Attempt (CFA) prediction is useful: predicting a grade from a student’s behavior and a given network structure. The resulting graph of users will then take into account seeking and disseminating tendencies, as well as an individual’s expertise, and can be visualized for an instructor’s benefit too.
Chapter 4

Integrated and Individualized Courses

Learning Data Analytics and Social Learning Networks themselves develop methods to improve instructional quality in indirect ways, by providing instructors with analytics to better help their students and by encouraging students to better help their peers. As discussed in Chapter 1, a step beyond is to use these algorithms in systems that can enhance the instructional process directly, especially for retaining quality when scaling up.

One approach for improving learning that has gained momentum is the Adaptive Educational System (AES), which aims to automate the instructional process. AES have demonstrated the potential to improve learning outcomes over one-size-fits-all (OSFA) course delivery in traditional classroom settings [7], and hold the promise of allowing individualized instruction to scale with increasing enrollment.

At the core of an AES is generally a user model (UM), which is defined and continually updated based on a student’s interaction with the system [30]. The UM is used to assist student navigation through the material (i.e., navigation adaptation) and/or to modify the presentation of the material itself (i.e., presentation adaptation).
Figure 4.1: IIC delivery is supported with two systems: a student-facing player and an instructor-facing dashboard. Under proper machine learning methodology, the behavioral measurements collected with the player lead to useful analytics and individualization.

To date, Adaptive Educational Systems have largely relied on a student’s assessment performance as the signal for updating the UM (e.g., by accumulating his/her points achieved on quizzes) and, in turn, for dictating which learning path the student will be assigned at a given time.

The potential of online learning platforms to integrate multiple learning modes – such as video, audio, text, slides, and graphics – into a course, and to collect behavioral data as students interact with these modes, presents novel opportunities to define richer UMs that can drive more effective adaptation decisions. The behavioral data and associated analytics we studied in Chapters 2 and 3 (for students watching videos and for students interacting on forums, respectively) can be employed in such behavior-based user modeling. Multi-modal integration is also an effective instructional style in its own right (i.e., aside from its potential to improve individualization), because it gives students the ability to choose which of the modes they prefer, and it provides increased opportunity for cognitive reinforcement from different perspectives [91, 95].

Motivated by this, we have worked together with UI designers and software developers in a substantial effort to create two systems – a student-facing course delivery
platform [26] and an instructor-facing analytics dashboard [24] which support the management and delivery of Integrated and Individualized Courses (IIC). The interaction between these systems and the engines driving them from the backend is depicted in Figure 4.1. Together with Zoomi Inc. [3], we have been working with educators, trainers, and tutors from different universities, corporations, and other institutions to run trials with their content in IIC format, testing conjectures about the efficacy of behavior-based individualization and the usability of behavioral analytics to instructors.

In this chapter, I will focus on the individualization framework (Section 4.2) and system architecture (Section 4.3) behind our AES, as well as two of the user trials (Section 4.4) that we have conducted with our AES using participants from our MOOCs. I will begin in the next section by describing related systems in the literature, to make the contribution of IIC clear.

### 4.1 AES in the Literature

Development of Adaptive Educational Systems dates back to the early 1990s. Brusilovsky presented a taxonomy and summary in 1996 [30]. We will discuss some of the well-cited AES that have been developed since then, and direct the reader to [30, 29, 7, 49] for more details.

#### 4.1.1 Navigation and Presentation Adaptation

ELM-ART [105] is a web-based AES which supports adaptive navigation through link annotation. The UM in ELM-ART is a multi-layered overlay, and is updated based on both knowledge inference from assessments and explicit user input. Our system is different in this regard because it also supports presentation adaptation, and because it does not allow instructors to directly modify the user model.
AHA! is another web-based adaptive system, where each page consists of a sequence of HTML fragments. Similar to our AES, AHA! supports both navigation (through link annotation and hiding) and presentation (through conditional inclusion of fragments) adaptation. The UM in AHA! is based entirely on a user’s browsing behavior, with fragments and pages being marked as desired or not based on pages visited previously. Our system instead uses assessments to infer user knowledge of and/or tendency towards learning concepts, with correlations with behavioral measurements to potentially enhance these inferences.

TANGOW also features navigation (through link disabling and adaptive link sorting) and presentation adaptation, but differently than AHA!, the HTML pages are generated dynamically at runtime from content fragments. As a result, the author must specify the sequencing of subtasks as well as the features of each fragment. One drawback to this approach (i.e., having no path generation, see Section) may be that the author must label each separate fragment, rather than starting with static content blocks and tagging the modifications. TANGOW allows storage of quiz scores and visited pages, leaving it to the author to decide if/how these will be used for adaptation.

CoMoLE is a Java-based AES that was built to support mobile delivery through a web browser, as opposed to our system which supports delivery through native app. It supports adaptive navigation by generating a list of recommended next activities, using (i) a rule-based filter which checks the context, features, and requirements of the activity against the UM, and (ii) a Markovian filter which analyzes learning paths followed by similar users/groups. The individualization framework proposed in is also based on user-to-user similarities, with promising results regarding student response from a system trial deployment. Our system is not currently focused on UM updates based on similar users, but rather based on fine-granular learning behavior students exhibit on content.
4.1.2 Adapting Learning Styles

Many AES have been designed to support adaptation based on a user’s inferred learning styles (LS) [29].

WHURLE [27] is an XML-based adaptive learning environment on which different user models can be instantiated. It supports adaptive presentation, by removing chunks of lessons that are not valid for the current user, but not adaptive navigation. Omission of a particular user model makes WHURLE a flexible system, but may add burden on the designer who must specify it [74]. The authors have evaluated WHURLE using two different dimensions of the Felder-Soloman Inventory of Learning Styles [52]: WHURLE-HM [27], with a UM based on the visual–verbal dimension, and DEUS [28], based instead on the sequential–global dimension, and surprisingly found no significant effect in favor of learning style-based adaptation.

LS-Plan [74] is a web-based AES with a user model based on four of the Felder and Silverman Learning and Teaching Style Dimensions [51]. This system supports adaptive navigation, and adapts by sequencing/re-sequencing the current learning path as opposed to our system which plans it one step at a time (see Section 4.2). The UM in LS-Plan is based heavily on assessment performance, but also uses lower and upper-bounds on the total time spent in a module to infer whether a user was on-task or not. Through experimentation, the authors found a statistically significant increase in the knowledge acquired from the adaptive modality.

4.1.3 Novelty of our System

We call our AES the Mobile Integrated and Individualized Course (MIIC). It possesses the following properties:

- It integrates video, text, assessment, and social learning into a single platform, and is thereby built for full course delivery.
• It captures clickstream measurements about students as they interact with the course material (including video-watching and pageview events), as they create notes, as they take assessments, and as they discuss socially.

• It can use the captured behavioral measurements to update the user model, and subsequently for individualization and for analytics.

• It can be delivered as a native mobile app, in addition to through a (mobile) web browser.

We are not aware of another Adaptive Educational System that integrates several modes, particularly lecture videos, likely because most have been focused on acting as supplements to traditional classrooms [99]. Also, other AES do not capture data on each mode at the level of granularity that our system does, and MIIC is the first to accomplish behavior-based individualization on learning data. Finally, AES that support mobile delivery via web browser have been developed [76], but none to our knowledge do so via native app.

To the last point on mobile app delivery, it is beneficial to have this option for a few reasons. For one, there has been a large increase in the popularity of tablet computers in the past several years, e.g., with global sales increasing by 50% in 2013 [53]. Also, studies have indicated that students may prefer learning on mobile devices than on PCs [39, 76]. Moreover, there have been a number of recent studies which have shown mobile device users to prefer apps to browsers for computing tasks (e.g., [108, 18]); in particular, [18] found this to be true in the context of accessing course resources. AES development in native app format has a number of implementation advantages, too, in terms of device-side storage, document pre-loading, and a wider range of sensors (i.e., camera and accelerometer) to detect user interaction which will be used in Section 4.3.
4.2 Individualization Framework

In this section, we will first present the general process we have following in designing our AES to-date. In doing so, we will discuss the options we have considered and have currently implemented for each of its four modules (see Figure 4.2). Subsequently, Section 4.2.2 will detail the individualization framework that was implemented at the time of the user trials, so that it is clear which portions led to the outcomes of these trials.

4.2.1 AES Design Process

Our AES design process consists of specifying four modules: inputs, user modeling, path generation, and path selection, as illustrated in Figure 4.2.

Inputs

This refers to the types of inputs that the AES collects. We identify four explicit types: assessment points, viewing behavior, social learning, and annotations. Additionally, pre-processing can be performed to give a richer and/or more useful set of inputs for the modeling stage. In particular, the algorithms for performance prediction that we studied as a form of analytics in Chapter 2 can be used to estimate a user’s score on assessments he/she did not take.

User Modeling

This module consists of machine learning techniques that map the inputs to update a low-dimensional user model (UM), which contains information about a student’s current state of learning [30]. We refer to the dimensions of the UM as the learning features of the course, which guide the content adaptation based on user knowledge and/or similarity to them. In a standard system, the feature set $\mathcal{F}$ will be author-
specified; the features can represent any of user “goals, knowledge, background, hyperspace experience, and preferences” [30]. We briefly discuss three possibilities:

(i) Learning styles: The author could designate the features to be different learning style preferences. These could be a subset of Felder and Silverman’s Learning and Teaching Style Dimensions: sensing–intuitive, visual–verbal, sequential–global, and active–reflective [51]. There are a number of other theories as well, such as those proposed by Dunn and Dunn [47] and Honey and Mumford [60].

(ii) Acquired knowledge: The author could also interpret features as dimensions of existing knowledge, covering key areas of the course. These would serve to track the knowledge acquired by the user while interacting with the course material, and could be very general in nature (e.g., “mathematical”, “conceptual”), or more specific, even to the point of simply having one feature for each segment of material.

(iii) Domain background: Additionally, features could measure user background in the content domain, to indicate whether or not he/she satisfies prerequisites for certain sections of material.
Score tracking. A standard UM update is through a score tracking system, where each answer choice in an assessment is associated with a number of points (possibly binary) for one or more features. This approach is taken in numerous developed systems because tests are the “most reliable source of evidence that a user has learned a concept” [105]. But quiz performance only gives part of the learning experience; it can be combined with behavioral data to gain a richer perspective of a student’s state [19]. This is particularly useful in scenarios like MOOC where assessments are not compulsory and tend to be a sparse source of data, as we saw in Chapter 2.

Behavioral correlations. With the assessments mapped to the features already, one way to incorporate other forms of inputs is through a large regression/classification problem that computes correlations among behaviors and scores. Consider, for example, representing video-watching behaviors as the sequences and/or aggregate quantities proposed in Sections 2.2 and 2.3 and determining the correlations with Correct on First Attempt (CFA) accordingly. Then, when the statistically significant behaviors are detected (e.g., a revising motif is triggered, the completion rate falls in a certain interval, or a certain number of visits are made to a specific chunk of a video), they can be mapped to updating the UM based on the correlations. An example of this is shown in Figure 4.3; this is the method used in many of the current deployments of the AES.

Another way of combining several input types would be to use a method like Factorization Machines (FM) [88]. In FM, each user-video pair is represented as a vector, say $x^k \in \mathbb{R}^D$ for pair $k$. The set of dimensions $D$ contains all the possible attributes of the pair, which can take binary values, or real values, such as the percentage of the video the user completed. Factorization Machines has been applied to educational data previously [100].

Automated feature extraction. We have also experimented with methods for extracting the feature set $\mathcal{F}$ when it has not already been specified. Matrix Factor-
Figure 4.3: A small example of using motifs, in addition to scores, to update the user model. Here, the UM has four dimensions: (i) revising on topic 1, (ii) scores on topic 1, (iii) revising on topic 2, and (iv) performance on topic 2. When a user watches a video containing these topics and takes a corresponding quiz, he/she revises on both topics and gets the quiz wrong. Her UM is changed accordingly (assuming quiz points and revisions have the same effect on the update for simplicity).

Path Generation

The purpose of this module is to specify each of the learning paths a user may follow as a result of the adaptation logic. This logic will compare the user model to the
properties of each path and select the one that best suits the user. We say that each learning path consists of a sequence of *segments*; one can think of a segment as the smallest unit of knowledge presented before/after an assessment. A segment may also have a number of different versions, corresponding to alternate presentations of the content. As such, we let \((s, v)\) refer to version \(v\) of segment \(s\), but we will only use the ordered pair when it is necessary to distinguish between versions. Then,

\[
S_u = ((s, v)_1, ..., (s, v)_n)_u
\]  

(4.1)

denotes user \(u\)'s learning path, which is the sequence of segment-versions (seg-vers) that he/she has visited.

For illustration, we can view a course as an author-defined network, where the nodes are segments (with different versions) and the links are potential transitions between them. In Figure 4.4(a), we show an example with 7 segments, where a link from \(s\) to \(t\) means that it is possible to transition to \(t\) once having finished with \(s\). Shown is an example learning path \(S_u = (1, (2, 1), (4, 2), 7)\).

It is important to note the difference here between how navigation and presentation adaptation \([30]\) are handled in our framework, which occur at the link level (e.g., direct guidance or annotation) and content level (e.g., collapsing/expanding or text emphasis), respectively. Navigation between segments encapsulates the former, while the choice of different versions refers to the latter.

Hence, for path generation, it is necessary to

(i) divide the content into segments (with different versions),

(ii) generate the set \(S\) of learning paths, and

(iii) specify the properties of the paths in terms of the learning features \(F\).
(a) Example lecture consisting of 7 segments, each with a different number of versions. Directional links denote potential transitions between segments.

\[ C_{st} = \{[\alpha, \beta], \forall f \in \mathcal{F}\}_{st} \]

(b) Illustration of transition terminology. Link constraints are specified by feature to limit the feasible region. Similar specifications are made for each version.

Figure 4.4: Diagrams to illustrate the definitions of learning path, segment-versions (seg-vers), and navigational and content adaptation.

For the trials in Section 4.4 we each perform of these manually, as will be explained. Other methods could automate a portion of this process for a given course. For example, if an author has completed (i), an instructor could then recruit a set of users to interact with the content, monitoring their satisfaction and progress as they make their own adaptation decisions. Based on the paths chosen by the users who
learned well, these actions could be hard-coded as paths for future users who have similar UMs at the decision points, thereby specifying (ii) and (iii). An alternative to these methods altogether may be to have no set paths at all, by having segments of content generate dynamically from content fragments based on the current user model, as proposed in TANGOW [33].

Path Selection

The last module is the method to select the learning path for each user based on the UM. In a static regime, the path is fixed based on information acquired at the beginning [27]. Our AES currently uses a step-by-step approach where the next segment is determined at the end of the current one, so only the learning path up to the current point is known. Another alternative is sequencing/re-sequencing, as with LS-Plan [74], where at any given point a user is assigned to an end-to-end path, which will switch if another is found more suitable to the current UM.

4.2.2 Individualization for User Trials

The individualization framework of our system at the time of the user trials in Section 4.4 consisted of the subset of the AES design options bolded in Figure 4.2. In our work with Zoomi Inc. [3], we have since extended it to include several of the other options – especially those in the modeling and path generation parts – as discussed above.

In particular, at the time of the trials, we had implemented three components: behavioral measurements, data analytics, and content/presentation adaptation.

Behavioral Measurements

As users interact with the MIIC application, their behavior is monitored and subsequently uploaded to a server. The following measurements were collected:
Figure 4.5: Illustration of how the different learning modes were integrated into the tablet version of MIIC. In (a), we show video and corresponding text, along with the ability to select text and add notes, perform external searches, and so on; in (b), an assessment is presented to the user; in (c), the video and text bookmark menu is selected, and a bookmark icon is on the page; and in (d), a note created by another user is selected on this user’s tablet.
**Viewing behavior.** Viewing measurements were taken for video and for pages as a whole. A snapshot of the user interface for the different learning modes is given in Figure 4.5(a). The clickstream measurements on the video (discussed in the context of Coursera in Section 2.3.1 and working the same, for all practical purposes, in our system), and the tags of the objects in the current page, were recorded with each touchscreen interaction. The interaction recorder that obtained these two types of viewing measurements will be explained in Section 4.3.

**Quiz responses.** Each quiz in our system was a series of radio-response multiple choice questions. Figure 4.5(b) depicts the standard assessment view that was shown. Each time a user answered a question, his/her response was recorded.

**Notes and markings.** MIIC allowed the users to take and share notes in their social network, as well as to place bookmarks on reading pages and in videos. Shown in Figure 4.5(c) is the user menu during the trials for video and text bookmarks, and in (d) is the note sharing aspect: the user can select a note made by another user in his/her social network and expand it to see it in full.

**Data Analytics**

We will present the method that was implemented for updating the UM as we define terminology:

**Feature weights.** Each segment $s$ was associated with a set of learning features $\mathcal{F}_s \subset \mathcal{F}$. The assessments within a segment serve to test user proficiency with one or more features $f \in \mathcal{F}_s$. Let $q \in Q_s$ denote question $q$ in the set of questions $Q_s$ for segment $s$. We refer to $w_{qf} \in \mathbb{R}$ as the weight of feature $f \in \mathcal{F}_s$ in $q$; if $f$ is not present then $w_{qf} = 0$.

**Assessment grade.** Let $c \in C_q$ be answer choice $c$ within the set of choices for $q$, and let $\pi_c \in \mathbb{R}$ be the points associated with choice $c$ in $q$. Upon completion of
segment $s$, the points awarded to a student for $f$ was given by:

$$\Pi_{sf} = \sum_{q \in Q_s} w_{qf} \left( \sum_{c \in C_q} \pi_c \times i_c \right), \tag{4.2}$$

where $i_c$ is 1 if choice $c$ was selected and 0 otherwise.

**Viewing ratio.** We refer to $d_s$ as the length of the video in segment $s$, and $e_s$ as the elapsed time spent by the user watching this video. The viewing ratio for a student in segment $s$ was then computed as $v_s = e_s/d_s$.

**Viewing grade.** This is denoted $p_v^f$ and gives the current grade obtained from tracking the viewing behavior for feature $f$.

**Points from completed segments.** This is a vector $a_c^f$ that contained the points awarded in all segments for $f$ in which the user had both viewed the video and taken the assessment.

**Viewing ratio from completed segments.** This is a vector $v_c^f$ that contained the viewing ratios for all segments for $f$ in which the user had both viewed the video and taken the assessment.

**Viewing ratio from watched segments.** This is a vector $v_w^f$ that contained the viewing ratios for all segments for $f$ that the user has watched.

**Correlation coefficient.** This, denoted $r_f$, is given by the sample Pearson correlation coefficient \[79\] between $a_c^f$ and $v_c^f$. This evaluates how similar/dissimilar the vectors are, and to what extent we can use $v_c^f$ as a measure of performance.

**Overall performance.** This is denoted $p_f$ and gave the overall performance model for the user on feature $f$. It was a smoothed version over segments of $g_f$, which is a convex combination of the two grades $p_a^f$ and $p_v^f$.

---

1 Note that this is the same as the fraction of time spent quantity defined in Section \[2.2\] 

2 Knowing in retrospect from the work in Section \[2.2\] that interval-specific correlations between fraction of time spent and CFA are more substantial, this can be readily modified according to those specifications in future work.
From these components, we parameterized the cumulative user data as \( \text{UD} = \langle \text{ac}_f, \text{vc}_f, \text{vw}_f \rangle \) and the user model as \( \text{UM} = \langle p^a_f, p^v_f, r_f, p_f \rangle \) \( \forall f \).

The UM update algorithm is given in Algorithm 3. It was called once the user exited the segment, but only if a value for either \( \Pi_{sf} \) (for any \( f \)) or \( \upsilon_s \) had been recorded; otherwise the user was be recommended to repeat the segment. \( \mathcal{F}_s \) is the set of features in which \( w_{sf} \neq 0 \); for all others, the model is not updated. When \( \Pi_{sf} \neq \text{NONE} \), this means the user received an assessment grade for \( f \), and \( g_f \) was updated accordingly. If \( \upsilon_s \neq \text{NONE} \) also, then the segment was completed, in which case \( r_f \) was updated. Following this, \( p^v_f \) was updated if \( \upsilon_s \neq \text{NONE} \). In this case, a new value of \( \upsilon_0 \) was computed based on the updated \( \upsilon^w_f \); \text{SCALE} subtracted the mean.
of \( \mathbf{v}_f \) from \( v_s \) to keep it relative to the user’s prior viewing behavior, and limited it based on \textbf{RANGE} to prevent \( v_0 \) from becoming excessively large. With \( v_0 \), \( p_f^v \) was updated. Finally, \( g_f \) was computed and \( p_f \) was updated. If the user had completed less than three segments for feature \( f \), then \( r_f \) would always be 1; in this case, we relied on the assessment grade for \( g_f \). Otherwise, \( g_f \) was a convex combination of \( p_f^a \) and \( p_f^v \) controlled by \( \gamma \in [0, 1] \). Subsequently, exponential smoothing controlled by \( \lambda \in [0, 1] \) was applied to \( p_f \) with the new \( g_f \).

\textbf{Intuition behind the viewing grade.} \( p_f^v \) was updated whenever a new \( v_s \) has been obtained, based on this ratio and \( r_f \). The method by which this occurred protected against updates when there is a lack of any trend. When \( r_f \) was close to +1, this indicated there was a strong, positive correlation between \( a_f \) and \( v_f \). In other words, there were many instances of the user performing better than average when spending more time with a video, as well as worse when spending less time. Then, if \( v_0 \) was positive the performance is bolstered, and if it was negative the performance was reduced accordingly. On the other hand, when \( r_f \) was close to −1, this indicated a strong, negative correlation, in which there were many instances of the user performing better than average when spending less time, as well as worse when spending more. Then, if \( v_0 \) was positive, the performance was reduced, and if negative it was increased. In any other case, the change to \( p_f^v \) would be negligible.

\textbf{Content/Presentation Adaptation}

In the rest of this section, we will explain how our system supported both navigation and presentation adaptation for the trials, referring to Figure 4.4(b) for terminology.

\textbf{Adaptive navigation.} Once the user finished working in a segment, our system would generate a recommendation as to which he/she should visit next. For the experiments presented in Section 4.4, we hard-coded the decision and brought the user to the next seg-ver automatically; this is a procedure known as adaptive hiding.
We have also experimented with a form of adaptive ordering, where the potential next segments are ordered based on relevance to the current UM, allowing the user to choose to follow the recommendation or not. We find that most instructional designers prefer the hiding approach because it makes the learning process appear seamless to the student.

Each link specified by the author would have constraints on the current \( p_f \). Letting \( F_{st} \) denote the set of features used to constrain the transition between \( s \) and \( t \), for each \( f \in F_{st} \) the author would specify a lower (\( \alpha_f \)) and upper (\( \beta_f \)) bound requiring \( p_f \in [\alpha_f, \beta_f] \) for \( t \) to be feasible. In Figure 4.4(b), these constraints are combined into a set \( C_{st} \).

Considering all potential transitions from \( s \), we obtain the set of recommended next segments as

\[
R_s = \{ t : p_f \in [\alpha_f, \beta_f] \forall f \in F_{st} \}. \tag{4.3}
\]

There are three cases:

- \( R_s = \emptyset \): This means that no transition is valid for the current user model. To avoid this problem, for each \( s \) the author designates one segment \( d_s \) to be the default transition from \( s \). In this case, then, the recommendation becomes \( \rho_s = d_s \).

- \( |R_s| = 1 \): Here, there is exactly one valid next segment \( t \) and \( \rho_s = t \).

- \( |R_s| > 1 \): The author should avoid this case by choosing a set of mutually exclusive constraints. If it arises, then the first valid segment \( u \) is chosen and \( \rho_s = u \).

**Adaptive presentation.** For each potential next segment, the most suitable version must be selected. The logic for this was very similar to (4.3): if \( F_{(t,v)} \) is the set of features used to define constraints for version \( v \) of segment \( t \), then the constraint for
to be feasible is that \( \rho_{j}^{f} \in [\alpha, \beta]_{f(\tau, v)} \), \( \forall f \in \mathcal{F}_{(t, v)} \). The difference from navigation is in terms of what is adapted. For a given segment, each version can have different properties in terms of content presentation, through the application of the following abilities of MIIC:

(i) Replacing: Based on the UM, specific pieces of content – videos, paragraphs, equations, or images – can be replaced with others. For instance, one version may contain more images and less text than another.

(ii) Collapsing/expanding: Content can also be collapsed or expanded. For struggling students this can be useful to elaborate on explanation details/revision and hide advanced material. For advanced students, elaborate explanations can be hidden.

(iii) Emphasizing: Content pertaining to learning features that a user possesses strengths/weaknesses in can be emphasized. For text, this includes modifying the font/color or highlighting. This helps a student to focus on these areas for reinforcement or improvement.

4.3 System Implementation

In this section, we will describe the architecture of the MIIC system for mobile app delivery that was in place during the trials. In particular, we will focus on the device and server-side implementation, as well as the interface between them. We will start with a description of the main components in each (Figure 4.6(a)), and subsequently will describe a flow chart that explains the process of selecting and rendering the next segment/version on the device (Figure 4.6(b)).
Figure 4.6: Architecture description of the MIIC implementation for the user trials. In (a), the key components coded and established for the user device and backend server are shown. In (b), high-level device-server interaction logic is shown, which is used to store user data, update the UM, fetch the next segment, and render it on the device.
4.3.1 Device-Side Implementation

Users accessed MIIC through a tablet computer. The main components we had designed for this device are illustrated at the top of Figure 4.6(a). Some of these resided in storage and others in memory.

**Device Memory**

The three main elements in device memory are the DPCM engine, interaction recorder, and user interface.

**DPCM engine.** Individualization must handle the process of dynamic content modification. Existing HTML rendering engines in mobile apps often rely on JavaScript, which, being an interpreted programming language, is too slow and inefficient for dynamic modification at this scale [44]. Issues with JavaScript become more severe in the case of platforms (such as iOS) that, for security reasons, disallow the use of Just-In-Time (JIT) compilation in third party applications. We saw this increased the execution time of JavaScript substantially.

The Dynamic Presentation and Content Modification (DPCM) engine in our system for mobile apps was instead a modified and optimized version of WebCore/WebKit. We extended WebCore with a C++ API to allow for native access to the Document Object Model (DOM) as well as to the layout engine for the different types of content modifications that MIIC performs (see Section 4.2.2). Additionally, we improved the SVG rendering library, which is used to display math equations. Due to the size of WebCore (roughly 10 million lines) as well as dependencies both within the library itself and with other frameworks, this required a few months of development.

**Interaction recorder (IR).** The IR monitored user interaction with the video player and with the content on each page as a whole. For the video player, the time interval between every two successive click actions – play, pause, jump, end of video, or close
app – was measured and recorded. The UNIX Epoch time, starting position, and interval duration were recorded in each case.

As for the page content, the time the user had spent viewing a page was recorded each time he/she switched the page or closed the app. We implemented a method to help check whether the user was viewing a page a given point in time. These took the form of four Boolean variables based on device sensors:

- **Last touch** ($TS_{PT}$): If a touchscreen interaction had occurred within the past $PT$ minutes, this was true. The last interaction time was kept in memory and updated whenever the view containing the text was touched.

- **Face detection** ($FD$): If the person’s face was detected in front of the device through the camera, this was true. The device pulled key frames from a continuous video stream to determine this.

- **Device angled** ($DA$): If the accelerometer detected that the device was held on an angle, this was true. This determined whether the user had the tablet flat on a surface by checking if the acceleration in any of the three dimensions of the standard Cartesian coordinate system differed from Earth’s gravitational acceleration.

- **Device movement** ($DM_{PD}$): If the accelerometer had detected device movement in the past $PD$ minutes, this was true. This checks if the user was holding the tablet.

Based on these, we defined another Boolean variable **viewing page** ($VP$) that was updated every 5 seconds. The following are the cases in which $VP$ was true:

(i) $TS_5 \land (FD \lor DM_1 \lor DA)$: If the user had touched the screen in the last 5 minutes, this is a good indication that he/she was focusing on the page. In addition to this, we

---

3Notice that this specification is similar to the clickstream measurements for Coursera described in Section 2.2.
required at least one of the other variables to be true to give some more evidence. For instance, if the user walked away from the tablet on the table, this condition would become false in quicker than in 5 minutes. We chose to not lower $P_T$ in case the user was reading a page without touching the screen.

(ii) $FD \land (DM_1 \lor DA)$: Expanding on the last point, even if the time since the last touch interaction had exceeded 5 minutes, the user may still have been viewing the text. What we required then was that they are in front of the tablet ($FD$) and that either of the accelerometer variables were true. Otherwise, it is likely they were sitting with the tablet in front of them but were engaging in some off-task behavior. Note that $DM_1 \land DA$ was not a sufficient condition; the tablet in this case could simply have been in transit.

Once the user switched the page, the UNIX Epoch time and counter duration were recorded as a pair. The counter duration measured time spent with any learning mode on the page, since $VP$ would be true in all cases. The set of text objects (i.e., each paragraph, image, equation, and heading) in the portion of the viewport that was currently visible was also recorded, to determine whether the page size was changed.

This IR logic was verified empirically prior to user trials. It was important to include it for data analytics, in order to reduce uncertainty associated with whether a user was currently on-task or not as will be seen in Section 4.4 when quantifying engagement in terms of page views. Distinguishing between student intents (i.e., their actual behavior) and their actions (i.e., their apparent behavior) is currently an active area of research for intelligent tutoring systems [34].

---

[3] User behavior captured by these types of sensors was not available with the datasets in Chapter 2 making it even harder to determine engagement. As mentioned, this is one of the advantages to mobile app learning technology.
Device Storage

The three main elements in device storage were course files, application, and behavioral data.

Course files. The text and image content of the course, as well as questions, were stored on the tablet in an EPUB container (video files, on the other hand, are streamed from the server-side). Each segment had its own UUID and was written as a separate XHTML file, and every object (i.e., paragraph, equation, heading, or image) was assigned a unique tag identifier. Different versions are created dynamically through applying tag logic to collapse/expand, replace, or highlight certain objects depending on the user model.

Additionally, each assessment question (with the answers) had its own XHTML file. The point allocations specified by the author were hard-coded into these files. A separate file containing the segment-question correspondence was also stored in the EPUB.

Application. This is the application itself, coded in the native language for the OS. It also included all settings for the app, including the user’s selections such as font size.

Behavioral data. The clickstream measurements (for page and video viewing in these trials) collected with the interaction recorder were written to storage each time a measurement was taken. Also, when the user answered question $q$ in segment $s$, the unweighted point values for $q$ were captured and stored.

4.3.2 Server-Side Implementation

A server running Apache was used for the backend. The main components are shown at the bottom of Figure 4.6(a); again, some resided in storage and others in memory. To communicate with the server, devices required an Internet connection, and submitted data using a REST API that sent HTTP POSTs with JSON objects as
the body. Server side code was written in Python with the Django framework and JavaScript.

Server Memory

The three main elements in server memory were the adaptation engine, video streaming, and annotation handling.

Adaptation engine. This engine had three functions: update the UM, determine the recommended next segment, and determine the potential next segments and versions (seg-vers). This corresponds to data analytics and content/presentation adaptation described in Section 4.2.2. All software for this engine was implemented in Python.

Video streaming. This implemented HTTP streaming to the native video player written in Java on the device. It received a URL request and fetched the corresponding video from storage on the server.

Annotation handling. For a given segment ID, the annotation handler fetched the necessary annotations from the user information database for the user. For bookmarks and highlights, this consisted of the positions at which the user had placed them, if he/she has used the segment previously. For notes, it fetched the position, content, and creator ID of any one falling into one of the following cases: (i) it was created by the user in the segment previously, or (ii) it was created by a user who has chosen to share it and who is socially connected to the user.

Server Storage

The three main elements in server storage were course database, user models, and user information. All of these were tables in an SQLite database.

Course database. This contained identification information about each course. All of this was added through a web interface written in HTML and JavaScript together
with a Python backend. The question-feature weights and segment-version transition logic for the course were stored here as strings.

**User data.** This is where the UM and behavioral measurements for each user were stored. Each entry specified the performance computation and interaction recordings for a user after he/she had completed a given segment.

**User information.** This contained other information about each user, such as their account, annotations, and social network (Facebook friends using the app).

### 4.3.3 Device-Server Interaction

The logic that was executed once the user had completed the current segment is outlined in Figure 4.6(b). First, the behavioral data collected with the interaction recorder was uploaded to the user data DB on the server, and any annotations made were uploaded to the user information DB. Then, the adaptation engine was fed with this, the UM, and the segment transition logic from the course DB. It returned an updated UM, $\mathcal{R}_s$, and the possible seg-ver pairs.

Once the selection was made (done automatically for these trials, but more generally could be driven by user input), the next seg-ver was fed to the video streamer, which would fetch the necessary video ID information from the course database and begin streaming to the device. Additionally, the annotation handling would check the user information DB for any markings the user has made in the segment previously, and would look at the social network identifiers of his/her “friends” (determined by friend sets on Facebook) to check for shared notes. Finally, the DPCM engine would render the content on the screen.
4.4 Two User Trials

Using a prototype of MIIC as an iOS mobile app, we conducted two user trials in 2013. Our objective was to evaluate the benefit that our system could bring to students in our Massive Open Online Courses (MOOCs).

4.4.1 Motivation: Heterogeneity in Our Courses

The challenge to obtaining effective learning at scale has been mentioned in various parts of this thesis. I am speaking here from experience. Recall that we have instructed two different MOOCs over several offerings since 2012; the data for two of these offerings was used in the second research thrust on Learning Data Analytics in Chapter 2.

The first one created was our undergraduate course *Networks: Friends, Money, and Bytes* ('FMB') [1], which was one of the six piloted by Princeton on Coursera in 2012. Technical in nature, it is oriented around 20 practical questions and introduces key conceptual advances and mathematical formulations along the way. During 2012-2013, almost 150K students enrolled in ‘FMB’ online, despite its calculus and linear algebra prerequisites and the lack of any certificates issued by Princeton.

Through a pre-course survey in the first offering of ‘FMB’, we made some interesting observations about the heterogeneous backgrounds of students. About half of the students were 30 years or older, and one quarter of the students did not have a college degree. Only 30% of the students were from the US, with 35% from Europe and Canada, and 35% elsewhere. Half of the students were taking this course as their first exposure to MOOC, and one third of the students were from backgrounds other than science, math, or engineering.

The diverse educational backgrounds became apparent to us through interacting with the students in the forums. We found that some were deterred because of
their lack of knowledge or interest in mathematics, while others were tackling advanced material. To cater to those deterred, we created a second course, *Networks Illustrated: Principles Without Calculus* (‘NI’) [2]. ‘NI’ is meant to explain the underlying concepts in ‘FMB’, but with much simpler mathematics. In offering ‘NI’, we then encountered the opposite problem: many students complaining that the material was too elementary, which caused some to lose interest.

### 4.4.2 Authoring Process

The experience with our courses described above is evidence of the heterogeneous learning needs and interests of the students in our MOOCs, and of those enrolled in massively scaled learning more generally. As a result, it was fitting to offer both of these courses together as a single, adaptive course through our platform, with the content delivered changing depending on the user model inferred for the student.

**Course Structures**

We converted two lectures from ‘FMB’ and ‘NI’ to MIIC, one per trial. The videos and assessments were taken from the respective courses, and the text from our books [41][23]. In architecting the features and transition logic, we set each MIIC lecture to present the most challenging content possible for each student, constrained by both his/her background knowledge in the prerequisite mathematics and his/her acquired knowledge at a given instant. An alternative would have been to set up these MIICs as intelligent tutors to bring everyone to the same level of understanding, by adapting the navigation through/around prerequisites. But without offering an incentive for participation, we decided to adapt to what students would want to learn. We did, however, structure each MIIC such that the key concepts were explained along any of the learning paths.
Figure 4.7: Course structure for the MIIC employed in our student trials.

Figure 4.7 shows the structure we employed for MIIC in both of the trials. Beginning with two separate books and courses, we decided to split navigation into two paths: segments at the top (2, 4, and 6) tended to contain content from ‘FMB’ and those on the bottom (3, 5, and 7) from ‘NI’, while the first segment was a combination and the last a summary. The number of versions shown for each segment here are specific to the second trial, though similar to the ones in the first.

User Modeling

For both trials, multiple choice questions were presented at the end of the segments. The 3 – 5 questions occurring at the end of each of segments 1 – 5 determined how the UM was updated. Each was tagged with up to three learning features: concepts (C), mathematics (M), and examples (E); hence $\mathcal{F} = \{C, M, E\}$. From (4.2) in Section 4.2, we specified the $w_{ql}$ as binary numbers, and the $\pi_c$ as integers between 0 and 3. For brevity, we omit the exact values for these for each trial, but note that segment 1 covered concepts $C$ and $M$, segments 2 and 3 covered $C$, $M$, and $E$, and segments 4 and 5 covered $M$ and $E$ in both cases. Also, for these trials, we chose to disable the viewing behavior portion of the UM update in Algorithm 3 for two reasons. First, we did not have video-watching data in advance to tune the parameters $\gamma$ and $\lambda$. Second,
we used these trials to collect such data for subsequent analysis and to establish a baseline of quality improvement to measure video-watching adaptation against.

Hence, the navigation decision in each trial upon completing segment 1 was dependent on $p_C^a$ and $p_M^a$. In subsequent segments, each version corresponded to content being emphasized (with color) or collapsed/expanded. Text tagged as corresponding to mathematics or concepts were colored depending on the current feature performance: green for high, and red for low. Additionally, in segments 4 and 5, intermediate steps in numerical examples were hidden depending on $p_M^a$ and $p_E^a$. And in segments 6 and 7, subsections corresponding to advanced material were expanded depending on the performance on all features.

The results of this modeling process, in terms of which learning paths were traversed, will be given for the second trial. The one-size-fits-all (OSFA) content in each trial consisted of content on the top navigation path in Figure 4.7 with no version modifications.

4.4.3 Trial 1: Student Response

The first trial was conducted in February 2013.

Research Questions

The purpose of this study was to investigate two questions:

- RQ1: Which features of MIIC are favorable among students, and which may need improvement?
- RQ2: How does student experience compare between MIIC and OSFA?
Figure 4.8: Illustration of a page in two different segments for MIIC in the first trial. (a) is a from segment 2, which shows a more in-depth treatment of the subject. (b) is from segment 3, which only uses basic algebra to explain the fundamentals.

Content

The content used here was a lecture on Google PageRank. For MIIC, Figure 4.8 shows an example of the difference in content shown on two different learning paths; referring to Figure 4.7, Figure 4.8(a) is from segment 2 and contains more advanced linear algebra, while Figure 4.8(b) is from segment 3 and explains the same features but only using basic algebra. OSFA in this trial was chosen to be a standard PDF version of the material, the implications of which will be discussed further below.
Procedure

We announced the trial for iPad users concurrent with the release of the lecture on Coursera. Since this was the first time the software and its backend were used by students, we wanted to ensure the initial infrastructure could readily support the scale of the trial, so we restricted participation to the first 100 students who responded to our first come first serve email. These users received a download link to both the MIIC (.ipa) and OSFA (.pdf) files. In order to reduce bias in the sequence of presentation, we divided them into two groups: one was instructed to use MIIC and then OSFA, and the other was to do the opposite.

Questionnaire

Upon completion of these tasks, each participant was asked to fill out a 14-question multiple choice questionnaire. Five questions asked about the perceived usefulness of the learning modes and overall experience with MIIC, for RQ1, and another four asked about the MIIC vs. OSFA comparison, for RQ2.

47 students filled out the questionnaire, and the 43 who indicated that they used both MIIC and OSFA are the focus of our analysis. This is much smaller than the MOOC enrollments cited in Chapter 1 because we limited participation by design. These sample sizes are on the same order as the size of traditional classrooms on which many AES have been tested [7]. Also, since OSFA was a PDF document in this trial, strictly speaking, the comparisons made here for RQ2 are between delivery with mobile, integration, and individualization versus delivery lacking these features. For this reason, we attempted to target most of the questions towards a single aspect of our design. Another approach may have been to make OSFA a multimedia eBook (i.e., MIIC without individualization), as is done in the second trial, though it is not clear whether students would prefer an eBook to a textbook (see e.g., [109]).
Results: MIIC Features (RQ1)

Lecture videos. One question asked about the usefulness of the integrated lecture videos. 68% of students found this very useful, 19% found it somewhat useful, and 13% found it not useful.

External search. Another asked about the usefulness of selecting text and searching it on external platforms. 36% and 38% found this very and somewhat useful, while the other 25% found it not useful.

Social notes. Another asked about the usefulness of being able to take and share notes. Only 23% and 28% found this somewhat and very useful, respectively, while the remaining 49% found it not useful. One possible reason for this is the limited time the participants had to interact in the trial.

Text emphasis. Another question asked how well the text emphasis helped to direct users to important concepts. 53% found this very helpful, 32% found it somewhat helpful, and only 15% found it not helpful.

Overall experience. Finally, one question asked how the student would rate the overall experience with MIIC, on a five-level Likert scale [74]. The distribution is shown in Figure 4.9: 38 (81%) responded excellent or good, and 9 (19%) responded moderate or poor.

Results: Comparing MIIC with OSFA (RQ2)

For each of these questions, participants were able to select (i) preference of MIIC, (ii) preference of OSFA, or (iii) indifference. A trinomial test described in [15] was used to determine whether there was a statistically significant difference for each question, using the number of positive (in favor of MIIC), neutral (no preference), and negative (in favor of OSFA) responses. The test statistic is the difference $N_+ - N_-$ between the number of positive and negative responses. Under the null hypothesis of no difference between the probabilities of positive or negative responses (i.e., $H_0 : p_+ = p_-$),
Figure 4.9: Student rating of overall experience with MIIC, on a five point scale. Over 80% responded “good” or “excellent.”

\[ N_+ - N_- \] has an expected value of 0 and a variance of \( 2np \), where \( n \) is the sample size (43), \( p = (1 - p_0)/2 \) is the presumed common value of \( p_+ \) and \( p_- \), and \( p_0 \) is the probability of an indifferent response. \( p \)-values were calculated using a normal approximation to the null distribution of \( N_+ - N_- \).

The four questions and their results, with significance evaluated at confidence levels of \( \alpha = 0.05 \) and 0.01, are as follows, and summarized in Table 4.1:

**Difficult material.** One of the questions asked which of the two contained excessive difficult material. 23 (53%) felt that each was fine, another 15 (35%) felt OSFA had too much, and 5 (12%) felt MIIC had too much. The \( p \)-value on this test was 0.025, significant in favor of MIIC at \( \alpha = 0.05 \).

**Simple material.** Another question asked which contained too much simple material. 29 (67%) felt each was fine, another 9 (21%) felt OSFA had too much, and 5 (11%) felt MIIC had too much. The \( p \)-value of 0.285 was not significant.

**Better understanding.** Another asked which of the two led to better understanding of the material. 26 (61%) were for MIIC, compared to only 10 (24%) for OSFA. The \( p \)-value of 0.008 was significant in favor of MIIC.
<table>
<thead>
<tr>
<th>Question</th>
<th>OSFA</th>
<th>Same</th>
<th>MIIC</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difficult Mat.</td>
<td>35%</td>
<td>53%</td>
<td>12%</td>
<td>0.025*</td>
</tr>
<tr>
<td>Simple Mat.</td>
<td>21%</td>
<td>67%</td>
<td>11%</td>
<td>0.285</td>
</tr>
<tr>
<td>Better Und.</td>
<td>24%</td>
<td>16%</td>
<td>61%</td>
<td>0.008**</td>
</tr>
<tr>
<td>Better Ov.</td>
<td>21%</td>
<td>9%</td>
<td>70%</td>
<td>&lt; 0.001**</td>
</tr>
</tbody>
</table>

Table 4.1: Results of the four survey questions asking respondents to compare MIIC and OSFA. Results tended in favor of MIIC: one was significant at a confidence level of $\alpha = 0.05$ (*), and two were significant at $\alpha = 0.01$ (**).

**Prefer overall.** The last asked which of the two the user preferred overall. 30 (70%) were in favor of MIIC, compared to only 9 (21%) for OSFA. The $p$-value was less than 0.001, significant in favor of MIIC.

### 4.4.4 Trial 2: Student Engagement

The second trial was conducted in September 2013.

**Research questions.** The purpose of this study was to investigate two more research questions:

- **RQ3**: Which learning paths do students of MIIC traverse as a result of the user modeling process?

- **RQ4**: Do students using MIIC have a higher level of engagement compared with those using OSFA?

**Procedure.** Three points distinguish the procedure of this trial from the first:

- (i) the content was Cellular Power Control;

- (ii) OSFA was given as an integrated mobile app, making the only difference between MIIC and OSFA the lack of adaptation; and

- (iii) each participant was only given either MIIC or OSFA, and was unaware of which he/she received.
Endpoints for engagement. In general, engagement is difficult to quantify, being defined as “the amount of physical and psychological energy that the student devotes to the academic experience”\textsuperscript{[10, 32]}\footnote{Recall the related discussion on measuring the “attention” received by a thread as the number of posts it received in Section 3.2.3}. In the end, we chose total page count as the main endpoint for engagement to investigate RQ4 (a similar endpoint was chosen in [106]). The reason for focusing on pages is two-fold, referring to the discussion in Section 4.3.1: (i) the pagecount timer captures the total time spent with any learning mode on the given page, and (ii) the IR logic helps reduce uncertainty in the recorded times. The fact that the same measurement is used for both MIIC and OSFA also helps to make the comparison more fair. The reason for using total count rather than time spent is that viewing for a longer time is ambiguous; it could mean higher engagement or more confusion. To account for differences arising from users changing page size, we used total object count a second endpoint.

Results: Learning paths (RQ3)

We will first give an overview of the learning paths traversed by the students, to give the direct results of the user modeling process outlined in Section 4.4.2. Here, we focus only on the users who were given MIIC, since OSFA had only a single path. Version encoding. Referring to Figure 4.7 different segment numbers were assigned the binary encoding given in Table 4.2 to describe the adaptive presentation. Each bit in the second column is a variable specific to the different sections:

- $C_{emph}$ and $M_{emph}$: These denote emphasis of conceptual and mathematical content, respectively. When set to 1, the color is green, and 0 means it is red.

- $M_{hid}$: This denotes hiding extra example steps. When set to 1, they are hidden, and when 0 they are not.
Table 4.2: Binary encoding of the version numbers for different segments in Figure 4.7.

<table>
<thead>
<tr>
<th>Segment(s)</th>
<th>Version Encoding (Base 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2&amp;3</td>
<td>(C\textsubscript{emph} \ M\textsubscript{emph})\textsubscript{2}</td>
</tr>
<tr>
<td>4&amp;5</td>
<td>(M\textsubscript{hid} C\textsubscript{emph} M\textsubscript{emph})\textsubscript{2}</td>
</tr>
<tr>
<td>6</td>
<td>(OL\textsubscript{exp} MR\textsubscript{exp} C\textsubscript{emph} M\textsubscript{emph})\textsubscript{2}</td>
</tr>
<tr>
<td>7</td>
<td>(SH\textsubscript{exp} OL\textsubscript{exp} C\textsubscript{emph} M\textsubscript{emph})\textsubscript{2}</td>
</tr>
</tbody>
</table>

- $OL_{exp}$, $MR_{exp}$, and $SH_{exp}$: These denote expanding the advanced material subsections “Open Loop Power Control,” “Matrix Representation,” and “Soft Handoffs.”. When set to 1, they are expanded, and when 0 they are collapsed.

Note that the reason the version counts in Figure 4.7 are not $2^x$ with $x$ the number of version variables is that some combinations are not possible.

**Analysis.** The 24 MIIC users who proceeded far enough to answer the questions at the end of seg 1 are the subject of our analysis here. Of them, 7 (29%) were navigated to segment 2 while 17 (71%) went to 3, meaning that the majority of students received the less difficult (‘NI’) path. 14 (58%) completed all questions on their respective paths, with a total of 9 distinct learning paths out of the 74 possible considering all combinations. These paths are shown in Table 4.3 along with the number of users for each.

The encoded variables range from a student who was navigated to the top path and had all variables 1 (third row) to a student who went to the bottom and had all variables 0 (fourth row). This corresponds to a range from the most to least advanced presentations possible, underscoring the heterogeneous demographic of the participants. The most common learning path was taken by four users (eighth row). While they were initially proficient in both concepts and math for their level ($C_{emph} = M_{emph} = 1$ in segment 3), as they moved through, they remained so in concepts ($C_{emph} = 1$ in segments 5 and 7) but began to struggle with math ($M_{hid} = 0$ in segment 5.
\[
\begin{array}{|c|c|}
\hline
\text{Learning Path} & \text{Frequency} \\
\hline
(2, (11)_2), (4, (000)_2), (6, (1000)_2) & 1 \\
(2, (11)_2), (4, (111)_2), (6, (1010)_2) & 1 \\
(2, (11)_2), (4, (111)_2), (6, (1111)_2) & 1 \\
(3, (00)_2), (5, (000)_2), (7, (0000)_2) & 1 \\
(3, (00)_2), (5, (010)_2), (7, (0010)_2) & 1 \\
(3, (10)_2), (5, (010)_2), (7, (0010)_2) & 2 \\
(3, (11)_2), (5, (000)_2), (7, (0010)_2) & 2 \\
(3, (11)_2), (5, (010)_2), (7, (0010)_2) & 4 \\
(3, (11)_2), (5, (011)_2), (7, (1111)_2) & 1 \\
\hline
\end{array}
\]

Table 4.3: Learning paths (omitting segments 1 and 8) among the 14 MIIC users who answered all the questions.

and \(M_{emph} = 0\) in segments 5 and 7). Additionally, no advanced material was shown to them \((SH_{exp} = OL_{exp} = 0\) in segment 7).

By including additional navigation paths in Figure 4.7, it may have been possible to split users who were on the same initial paths further. This may have been beneficial for those 70% who were initially navigated to segment 3. To investigate this, we considered the subset of this 70% that were on the borderline of navigation to segment 2 (roughly speaking, having achieved \(\geq 70\%\) of the required points). Only 5 of these 24 satisfied this criteria, which is too small to make statistical claims with, but it is surprising that of these 5, 3 completed the lecture. The 60% finishing rate of this group is roughly the same as the 58% rate of the 24 participants as a whole. This means that there was no evidence that having an additional navigation path for this group would have helped their completion rate.

Results: Page and object count (RQ4)

Data handling. Since many users did not traverse far into the lecture and a number of the entries constituted a short duration more in line with browsing than studying, two filters were created: (i) an entry in the database was only considered valid if the elapsed time was at least 10 seconds; and (ii) only users who reached segments
Figure 4.10: Distribution of the (a) pages and (b) objects accessed by students in the OSFA and MIIC groups. The distribution for MIIC is shifted to the right, indicating higher engagement (with the exception of one outlier).

2/3 were considered, since these were the users who experienced the effect or lack of adaptation. Combined with the IR logic, the 10 second cutoff was a second precaution taken to discount entries most likely associated with off-task browsing or skipping through the material. Including it was seen to help discount a number of users with this apparent behavior.

There were 44 users who satisfied these criteria: 25 in the MIIC group and 19 in the OSFA group. Figure 4.10 gives boxplots of the two endpoints by group. In (a), the mean (resp. standard deviation) for MIIC is 10.76 (resp. 5.95) pages, compared to 6.26 (resp. 4.92) for OSFA; in (b), these values are 74.12 (resp. 39.45) for MIIC compared to 46.68 (resp. 41.93) for OSFA. The distribution for MIIC is visibly shifted to the right, suggesting a higher level of engagement when quantified in terms of page and object counts.
Table 4.4: \( p \)-values, 95% confidence interval (CI), and Hodges-Lehmann estimate (HLE) of the difference between the MIIC and OSFA groups for the distributions in Figure 4.10(a) and (b), respectively, using the Wilcoxon test.

<table>
<thead>
<tr>
<th></th>
<th>Total page count</th>
<th>Total object count</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )-value</td>
<td>0.009**</td>
<td>0.015*</td>
</tr>
<tr>
<td>95% CI</td>
<td>(1.00, 8.00)</td>
<td>(7.00, 57.00)</td>
</tr>
<tr>
<td>HLE</td>
<td>5.00</td>
<td>30.00</td>
</tr>
</tbody>
</table>

**Analysis.** Since Shapiro-Wilk tests detected significant departures from normality, non-parametric tests were preferred over the standard t-test. The Wilcoxon Rank Sum (WRS) test [93] is a nonparametric procedure which is more sensitive to differences between central tendencies than others; as in other parts of this thesis, we therefore employed this as our primary method, with a continuity correction to the discrete distribution of the test statistic. Using this test, we computed (i) a two-sided \( p \)-value for testing the null hypothesis of no difference, with significance evaluated at \( \alpha = 0.01 \) or \( \alpha = 0.05 \), (ii) a 95% confidence interval (CI) estimate for the shift in location, and (iii) a Hodges-Lehmann estimate (HLE) of the shift; the HLE is an estimate of the shift in location parameter based on the WRS test.

The results are given in Table 4.4.

**Significance testing.** For pages, a \( p \)-value of 0.009 was obtained, significant in favor of MIIC. For objects, the \( p \)-value was slightly higher (0.015) but still significant.

**Confidence interval.** For pages, the difference between the means was between 1 and 8 with 95% confidence. For objects, it was between 7 and 57.

**Distribution shift.** The HLE of the difference for pages was 5, and 30 for objects. These shifts are large when considering the maximum counts from students in each case (23 pages, 153 objects). The percent increase in mean from OSFA to MIIC was 71.8% for pages 58.8% for objects.
4.4.5 Key Messages

The following are the key conclusions drawn from these two MIIC trials:

- MOOC students using MIIC tended to have higher engagement than those using OSFA, when quantified in terms of page counts.
- MOOC students responded favorably to most of the features of MIIC (e.g., lecture videos and text emphasis), but not to the social learning aspect.
- MOOC students favored course delivery via MIIC to OSFA on a few dimensions, including overall preference and better understanding.

4.5 Extensions

Our MIIC delivery platform is the first to enable behavior-based individualization, i.e., adapting based on a full range of behaviors rather than just performance. Still, these initial trials with our system were limited in scope, perhaps raising more interesting research questions than those answered. As a result of this, we are continuing to explore several next steps, which I will describe in the following paragraphs.

First and foremost is additional data analytics on IIC students, as was done on the clickstream data in Chapter 2 and on the social network data in Chapter 3. By analyzing the data collected from each separate learning mode in our course delivery player (rather than collectively, as is the case with total page count), each can serve as a different proxy of engagement. Together with Zoomi Inc. 3, we have been deploying IIC through several different channels to obtain data for this.

Related to this is trialling other courses and course types. The trials discussed here only include MOOC students from our own networking courses. In order to evaluate it in a more general setting, we are working with authors in different disciplines, from academic to training, to transform their content to IIC format and run further
trials. In working with additional authors, we will also change our endpoints to reflect the measure of efficacy in the given setting. For example, in a class with strict learning goals, we can treat incremental performance as a primary endpoint \cite{74}. In a corporate training course, the change in productivity (e.g., difference in number of trouble tickets solved by a technician before and after taking the course) can be used, provided that such data is made available.

More fundamentally, one question remaining is: \textit{How to quantify the benefit of behavior-based individualization over quiz-based adaptation?} To assess this, we plan to compare learning outcomes from two versions of a course: One employing full behavioral modeling, and one with just performance modeling. In conducting trials, students will be randomly allocated to receive one of these versions, and given the same pre-test and post-test. The difference in test scores then becomes the primary endpoint.

Related to this, another important extension is to run user trials with more advanced machine learning techniques for user modeling, \textit{i.e.}, beyond Algorithm \ref{alg:behavioral-modeling} now that we have a baseline for comparison and data to tune the algorithm parameters on the behavioral features in advance. As discussed in Section \ref{sec:behavioral-modeling}, our current method incorporates the motifs that were identified as being significantly correlated with quiz performance in Section \ref{sec:quiz-performance} for updating the user model and triggering different learning paths.

To this point, there are three potential benefits of having multiple measures of “performance.” First is that performance can be updated even if the user has chosen to skip an assessment, which will be particularly useful in a situation like MOOC where quiz responses may only be optional. Second is that with additional information, the effect of measurement noise associated with guessing correctly and slipping behavior (\textit{i.e.}, answering incorrectly when the user actually knows the information, see \textit{e.g.}, \cite{82} for a discussion) can be reduced. Third is that quiz scores may not be
an appropriate measure of performance, especially in a platform where learners have
the ability to return to an exam question several times before actually submitting
an answer; this encourages the behavior of skimming through the content in-between
quizzes to search for the correct answers, without actually studying the material.

Finally, as stated, enabling individualization in any AES requires instructors to
provide significant upfront input to establish the adaptation logic. In our platform,
this involves

- (i) defining learning topics,
- (ii) labeling content with these topics, and
- (iii) specifying decision points and thresholds.

We have therefore been asking: *How can we reduce the amount of input an instructor
must provide to an AES?* One of our approaches involves automating (i) and (ii)
through topic extraction from the constituent words in each learning mode, as de-
scribed briefly in Section 4.2.1. With this information, a number of possible decisions
can become defaults for (iii), *e.g.*, upon detecting a user struggling, route him/her
back to previous content with the same topics. Another of our approaches will lever-
age prior learner behavior: for a given course, users will initially run through the
non-adaptive version, with their behaviors collected. A variety of paths that have
successful outcomes when taken would be identified, and the behaviors exhibited
by students on those paths would define associated learning styles. Future students
would be routed according to the successful path most closely matching their observed
behavior.
Chapter 5

Conclusion

Recent innovations in communication technology have made it feasible to scale up education and learning. We have witnessed up to hundreds of thousands of students enrolled in single Massive Open Online Course (MOOC) offerings today, a truly remarkable milestone for learning technology that holds the promise of global access to world class instruction eventually. Before that is possible, however, several challenges to enabling effective learning at scale, further complicated by heterogeneity and asynchrony, must be overcome. These challenges present substantial opportunities for researchers in engineering, computer science, learning sciences, education, and other fields to work together in defining the next generation of learning technology systems.

In this thesis, I have described many of the research questions – some opened and some addressed to varying degrees – towards scaling the quality of learning that is obtained in traditional classrooms to online learning scenarios. In particular, I detailed our investigation into several of these questions, divided into three main thrusts: Learning Data Analytics (LDA), Social Learning Networks (SLN), and Integrated and Individualized Courses (IIC). In LDA (Chapter 2), we designed new ways of representing learner video-watching behavior, and applied these representa-
tions to enhancing performance prediction for early detection of struggling/advanced students. Then, in SLN (Chapter 3), we proposed a model for peer-based learning to evaluate the efficiency of discussion forum interactions, leading to algorithms for encouraging more effective social learning. Finally, in IIC (Chapter 4), we designed a framework for accomplishing behavior-based individualization, implemented this in our own learning technology platform, and described the results from the preliminary user trials conducted with this platform. We saw too how the algorithms from LDA and SLN can be used for user modeling in the IIC individualization framework and for instructor analytics.

More generally, the work on LDA, SLN, and IIC that I presented are important steps in the development of systems to help students, to help instructors better help their students, and to help students better help each other in online learning scenarios, both MOOC and otherwise. They are part of a burgeoning research area in learning quality enhancement, the principle components of that field being recommendation (e.g., of content modules to visit), prediction (e.g., of future performance), incentivization (e.g., of student participation), visualization (e.g., of social learning networks), integration (e.g., of multiple learning modes), and individualization (e.g., of course delivery). Many of the remaining research questions require large-scale deployment of systems with these methods in place, to verify their usability and quantify their impact on learning efficacy.

To this end, we have been working with Zoomi Inc. on incorporating our algorithms into the company’s product suite – consisting of a course delivery player, instructor analytics dashboard, and individualization engine – so that we can obtain broad sets of data from different learning scenarios and test conjectures about behavioral analytics and behavior-based individualization. It will be particularly interesting to see how the correlations between behavior and performance, the efficiency of existing social learning networks, and impact of individualization that we investigated
for MOOC in this thesis change depending on the type of learning scenario being considered.

It is an understatement to say that it will be interesting to see what the next decade – and beyond – has in store for education and learning innovation.
Bibliography


[34] Suleyman Cetintas, Luo Si, Yan Ping Xin, and Casey Hord. Automatic Detection of Off-Task Behaviors in Intelligent Tutoring Systems with Machine


[86] Jiezhong Qiu, Jie Tang, Tracy Xiao Liu, Jie Gong, Chenhui Zhang, Qian Zhang, and Yufei Xue. Modeling and Predicting Learning Behavior in MOOCs. In Conference on Web Search and Data Mining (WSDM), 2016.


