UNPACKING THE EDUCATION PRODUCTION FUNCTION: ESSAYS ON TEACHING, PARENTING, AND MEASUREMENT IN EMERGING CONTEXTS

A DISSERTATION
SUBMITTED TO THE GRADUATE SCHOOL OF EDUCATION
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Zhaolei Shi

June 2021

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Eric Bettinger) Principal Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Susanna Loeb) Principal Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Susan Athey)

Approved for the Stanford University Committee on Graduate Studies

Preface

This dissertation consists of three essays that investigates topics concerning the three main characters in the education production function – the teacher, the parent and the student. All three essays speak to modes of learning that have very broad reach. Mobile education in the cases of papers one and three, higher education in developing countries in the case of paper two. All three papers leverage novel sources of information to inform policymaking, design, and future research. Papers one and two are empirical inquiries powered by experimental and quasi-experimental evidence respectively, while paper three expands the methodological toolkit available to researchers and practitioners.

The first chapter of this dissertation brings experimental evidence to the problem of improving parental follow through in mobile education. The paper is authored by me while its contents belong to a part of a larger project that will be published in co-authorship with Susan Athey. Mobile learning apps offer us the opportunity to improve parenting at an unprecedented scale around the world. Effective parenting on these platforms is constrained by follow through. I investigate whether features on a recommendation page can increase follow through. I find that Cognitive overload is a major factor affecting parental follow through. Providing a top-of-the-page reason significantly decreased the completion rate by 70%. We also find null effects on completion from adding 1) a commitment question, 2) a guide to navigate back to the page, and 3) a link to the child's learning report. The hypothesized positive effects from these features are likely negated by the cognitive load penalties. However, increasing the number of recommended modules increased the number of modules each complier completed without increasing cognitive overload.

The second chapter of this dissertation presents quasi-experimental evidence for an age-old question – how does faculty research affect student learning? The paper is co-authored with Prashant

Loyalka. Other co-authors who contributed to data collection and discussions include Guirong Li, Elena Kardanova, Igor Chirikov, Ningning Yu, Shangfeng Hu, Huan Wang, Liping Ma, Fei Guo, Ou Lydia Liu, Ashutosh Bhuradia, Saurabh Khanna, and Yanyan Li. Whether faculty research affects college student achievement has long been the subject of debate. Previous studies use subjective measures of student achievement, focus on correlation rather than causation, and typically focus on one college or department, thus lacking generalizability. Using unique, large-scale survey and assessment data that we collected from nationally representative samples of four-year STEM undergraduates and faculty in China, India, and Russia as well as a student fixed effects identification strategy that accounts for differential sorting of students to faculty, we present generalizable estimates of the effect of faculty research on objective, standardized measures of student achievement. Results show that faculty research has a negative effect on student achievement, suggesting direct tradeoffs between the university’ s dual mission of producing research and learning.

The third chapter of this dissertation concerns the estimation of student models using large sparse data from online learning. The paper is authored by me while its contents belong to a part of a larger project that will be published in co-authorship with Susan Athey. The rise of popular mobile education applications produced data where a large number of students each answers a small subset of questions from a large question bank. Traditional approaches from the education measurement literature face important limitations in this context where data is large but sparse. We propose models based on latent factorization and Bayesian variational inference to address these challenges. Our models retrieve true parameters with greater fidelity than traditional models in simulations. They also scale well computationally to industrial-size datasets. Compared to traditional specifications, latent factorization models can make more accurate predictions on the hold-out test set in general. More latent factors and adding hierarchical dependence on question attributes contribute to better predictive performance in lower-frequency content areas. Our models also compare favorably in both computational performance and predictive accuracy against similar models from recent literature. We conclude by describing a real-world application of our models in personalizing homework assignments. In a future study, we plan to run experiments with this application to quantify the impact of personalization.

Contents

List of Tables

List of Figures

Chapter 1

Improving Parental Follow Through in Mobile Education

1.1 Introduction

Large-scale mobile learning apps have risen in popularity in recent years. Byju's in India, 17Zuoye in China, and Khan Academy in the US are just some examples of companies that offer significant mobile options. Together, these three companies account for more than 180 million users worldwide (Singh, 2020; Khan Academy, 2020a; Sunny Education Inc., 2018). These apps often offer features for parents to enhance their engagement with their children's learning. Features for parents include those used to monitoring learning, increase practice and exposure, as well as personalized recommendations for shoring up weak links.

Mobile learning apps offer us the opportunity to improve parenting at an unprecedented scale around the world. This holds great promise for education in general, since parents are integral in their children's life successes. However, several key obstacles stand in the way of realizing this vision. Unlike traditional venues of parental engagement (e.g. teacher-parent meetings), effective parental engagement require three key components (Figure 1.1).

Firstly, parents need to use the App. With the explosion of number in the number of apps we install on our phones, this is not an easy task. Secondly, education apps are typically overflowing

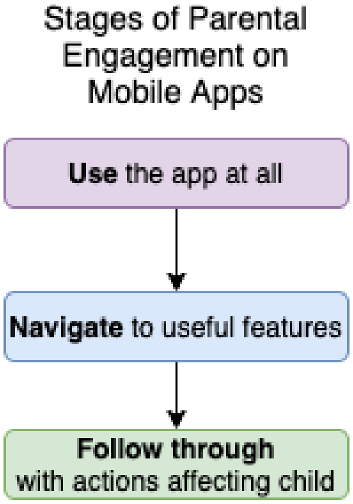

Stages of Parental Engagement on Mobile Apps
Use the app at all
Navigate to useful features
Follow through with actions affecting child

Figure 1.1: Effective parental engagement in mobile Apps

with complex features, it is hard for the parent to navigate to useful features most useful for their child's learning. Thirdly, even when a useful course of action is identified, the parent still needs to follow through and take action with their child to generate any meaningful impact on her.

In reality, mobile platforms face challenges in engaging parents in each of the three stages named in Figure 1.1. Even with a large user base, the average parental usage rate can be very low. Apps have many complex features and parents have a hard time navigating to useful content. Finally, parents often do not follow through in taking action with their children.

This study is a part of a series of studies examining each of these issues through experimentation. In this paper, we seek to address the issue of parental follow through with a large-scale experiment leveraging insights from behavioral economics. Behavioral interventions in parenting have been increasingly studied in recent years (see Bergman (2019) for a review).

In this study, we randomized the features shown to parents on a page recommending customized Learn-Practice-Explain (LPE) exercise modules for their children. We find that cognitive overload is a major factor in our experiment with features shown on a recommendation page. Contrary to our hypotheses, we find that providing a reason for recommendation at the top of the page significantly decreased the completion rate of LPE modules by 70%. Consistent with a cognitive overload interpretation, users who were shown a reason were 60% less likely to answer the survey question, but those who responded provided statistically indistinguishable answers from the control group. We also find null effects on completion from adding 1) a commitment question, 2) a guide to navigate back to the page, and 3) a link to the child's learning report. The hypothesized positive effects from these features were likely muted or negated by the penalties they induce as cognitive overload.

While we find strong evidence of cognitive overload from the above-mentioned features, we find no evidence that increasing the number of modules increases cognitive overload as users shown more modules were equally likely to answer the survey question. Increasing the number of recommended modules did not increase the number of users completing any modules, but it increased the number of modules each complier completed.

Our experiments advance the existing behavioral literature in parenting through observing fine-grained behavioral data generated in mobile apps. We also reveal a set of issues in parental follow through that can inform future researchers and practitioners. Finally, my results contribute to the understanding and improvement of an important mode of education that serves a large number of the world's learners. The paper is organized as follows. Section 2 samples the related literature, section 3 describes our partnering company's platform, section 4 details the experimental design, section 5 lays out our results, and section 6 concludes.

1.2 Related work

Our study builds on an expanding literature of behavioral experiments in parenting (see Bergman (2019) for a review). Many randomized studies have revealed the salience of behavioral barriers may prevent parents from investing optimally in their children's education. Among the barriers identified are biased beliefs about their children's performance, limited cognitive bandwidth, and the cost of monitoring. Interventions ranging from text messaging about student's behavior (Bergman, 2015) to tax assistants (Bettinger et al., 2012) have been shown to be effective at addressing one or multiple barriers.

Apart from their primary findings, these studies often also uncover non-obvious pitfalls. Cunha et al. (2017) found that interactivity in a message program decreased learning outcomes. In addition, Gallego et al. (2017) found that the lack of certain cues was responsible for a loss of half of the effect of the program. Bergman et al. (2019) found that opt-in programs fail to deliver an impact due to very low uptake rates (less than 11%) compared to an opt-out program (95% uptake).

Our study is related to the growing literature on personalized learning. Education technology companies have long made claims about the imperative of personalization. Efficacious programs, such as those examined by Muralidharan et al. (2019) and Banerjee et al. (2007), have personalized components. However, other technology-based learning programs without personalization have shown to be similarly efficacious (e.g. Mo et al. (2014); Lai et al. (2012)). We contribute to the literature on personalization by providing experimental evidence on a recommendation page for parents filled with personalized content for their children.

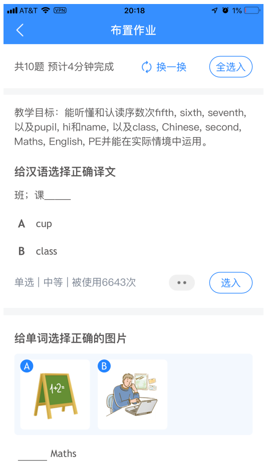
Since our experiment takes place on a recommendation page, we are also inspired by the literature around the design of recommendation systems. For example, in the literature on features of recommendation systems, user experience studies found that a larger number of recommendations can introduce choice fatigue (Bollen et al., 2010; Konstan and Riedl, 2012). Pu and Chen (2006) documents how explanatory text increase trust in recommendation system and intention to return to the page. We run multiple interventions in our experiment and one of our interventions adds an explanation to the page explaining to the parent why the exercise modules were recommended to her.

Our study is also related to the behavioral literature on commitment devices. Commitment devices have been studied extensively in behavioral economics. Bryan et al. (2010) discusses the various theories on commitment devices. They highlight the difference between hard and soft commitment. The latter incurs psychological costs instead of real economic penalties. Soft commitment devices have been applied to financial discipline (e.g. Thaler and Benartzi (2004)) and charitable giving (e.g. Breman (2011)). Recently, Mayer et al. (2015) reports on using commitment devices on parents to increase their usage of a reading App. One of our interventions involves varying the presence of a soft commitment question that parents can choose to check.

Finally, our study is closely linked to the literature on technology-induced cognitive overload. Originally proposed to explain the missing link between IT investment and productivity gain (Karr-Wisniewski and Lu, 2010), technology-induced cognitive overload has been observed in many settings beyond the workplace. In educational settings, teachers who taught classes that had more complicated reports generated by an AI tutor used the reports less and, in turn, had much smaller impacts on their student achievements (Kim et al., 2019). Information overload patterns have also been observed with parents. For example, (Fricke et al., 2018) finds complicated messages increased drop out from the text messaging program. Cortes et al. (2018) found that compared to 3 messages per week, 5 messages per week decreased the effect of a parental messaging experiment.

1.3 Context

Our study takes place on the platform of our partner company, 17Zuoye. 17Zuoye runs 3 separate apps for elementary school teachers, students, and parents in China. They enjoy a high level of adoption in the country with 60 million users in 2018 (Sunny Education Inc., 2018). The app is designed around the functionality that allows teachers to post homework to students. The draw for teachers is that the app provides them with ready-to-use homework questions and grades their students' responses automatically, reducing their workload in day-to-day teaching. The app is free to use for teachers, students, and parents. For their revenue stream, 17Zuoye runs a "freemium" model for parents who want to purchase education products in the App.



共10题 预计4分钟完成

教学目标：能听懂和认读序数次fifth, sixth, seventh, 以及pupil, hi和name, 以及class, Chinese, second, Maths, English, PE并能在实际情境中运用。

给汉语选择正确译文

班；课_____

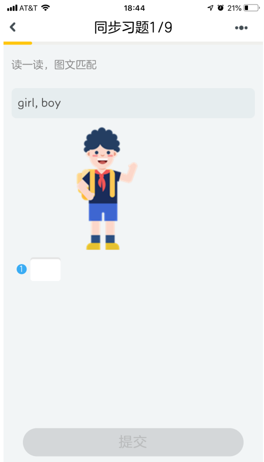
A cup

B class

单选 | 中等 | 被使用6643次

给单词选择正确的图片

_____ Maths



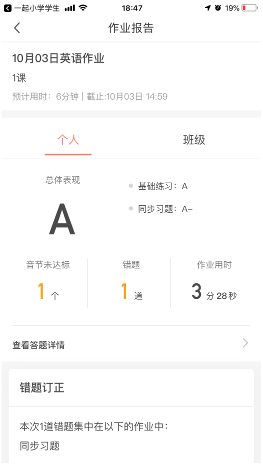
读一读，图文匹配

girl, boy

Figure 1.2: Assigning homework in the teacher app

Figure 1.3: Completing homework in the student app

Figure 1.2 shows the teacher app's interface for assigning homework. Teachers get to choose from a bank of curriculum-relevant homework questions to assign to their class. Students complete the assigned homework on the student app (see Figure 1.3). Their responses are automatically graded and recorded in the system. Teachers have access to student records through a data console in the teacher app.



一起小学学生 18:47 19%

作业报告

10月03日英语作业

1课

预计用时：6分钟 | 截止:10月03日 14:59

个人 班级

总体表现

A

基础练习：A

同步习题：A-

音节未达标 1 个

错题 1 道

作业用时 3 分 28 秒

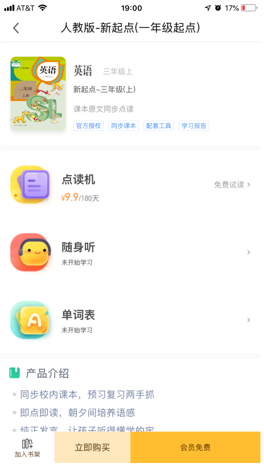
查看答题详情

错题订正

本次1道错题集中在以下的作业中：

同步习题

Figure 1.4: HW monitoring page in the parent app



AT&T 19:00 17%

人教版-新起点(一年级起点)

英语 三年级上

新起点-三年级(上)

课本原文同步点读

官方授权 同步课本 配套工具 学习报告

点读机 ¥9.9/180天 免费试读 >

随身听 未开始学习

单词表 未开始学习

产品介绍

- 同步校内课本，预习复习两手抓
- 即点即读，朝夕间培养语感
- 纯正发音，让孩子听得懂学的宏

加入书架 立即购买 会员免费

Figure 1.5: Purchase page in the parent app

In the parent app, parents have access to their child's homework records. Parents get to monitor their child's homework performance on particular pages where this information is displayed. Parents can also purchase education products for their children. Examples of products include educational games for math designed as a jungle adventure, e-books designed as a recap of course material, and online oral English courses taught in real-time. The app also carries a wide selection of free study resources such as English videos, ebooks, and practice questions. Parents may spend time in the app browsing these free items and assigning them to their children if they choose to.

1.4 Experimental designs

In the recommendation page features experiment, we want to investigate how features impact parental satisfaction and follow-through. To do this, we randomized the features shown on a page recommending customized Learn-Practice-Explain (LPE) exercise modules for their children. While the platform has content for math, English, and Chinese, LPE modules were only available for math. The recommendation page was only available to parents whose child has completed a math exam on the platform. The recommendation page is designed to prompt parents to get their child to complete the LPE exercise using the parent's cell phone.

Recommended modules LPE modules are generated based on the questions students missed on the math exam. Questions on the math exam are associated with knowledge points. Each LPE module is specially designed to teach one knowledge point (e.g. ”interpret a figure to write down a multiplication expression”). For each student, her exam scores would be aggregated into the correct rate by each knowledge point. The LPE modules would then be ranked in order from the knowledge point with the lowest correct rate to the knowledge point with the highest correct rate. If the student received a perfect score on the math exam, then no LPE module will be recommended to the student. In this case, the parent does not have access to the recommendation page.

Our experiment population is composed of 51860 parents who entered the recommendation page between Jan 14th, 2021, and Jan 29th, 2021. Parents entered the page through either the entry point on the exam report page in the App or by clicking a system-generated push message reminding them to open the recommendation page. Our treatment conditions are variations of the features shown on the recommendation page. Assignment to treatment arms is randomly assigned and orthogonal to how the parent entered the page.

1.4.1 Sample recruitment and data collection

Sample recruitment took place on 17Zuoye's platform in their parent app. The recommendation page was available to any parent whose child had taken a commissioned exam. The parents who chose to open the recommendation page when the experiment was ongoing were all assigned to one of the treatment conditions. While our interventions were directed at parents, they call on parents to take action with their children by getting them to complete exercises on the parents' phones. Thus, the assignment to treatment conditions may affect children's outcomes as well. Data collection is done automatically through 17Zuoye's data infrastructure.

1.4.2 Learn-Practice-Explain Module

The LPE modules integrate animation with practice questions and explanatory voice-over and text. The student is guided through a series of questions and animations that explains one knowledge point (see Figure 3.17). The typical time to complete an LPE module is between 2 and 4 minutes. As the student answers the questions, only a correct answer would allow the student to move forward. A wrong answer would trigger a hint and the student is directed to answer the question again. Importantly, the modules are designed to target granular knowledge points. For example, "rewriting numbers in units of Wan (ten thousand)" is a knowledge point and correspondingly, an LPE module, in 4th-grade math.

1.4.3 Sampling and treatment assignment

Our experimental population comes from elementary school parents whose children partook in math subject exams on the 17Zuoye platform. These exams are created by teachers, schools, or the local education authority and delivered to students via 17Zuoye's app. Exam questions are labeled according to a knowledge framework. As such, our recommendation page generates recommended LPE modules that tackle the same node in the knowledge framework as the students' most error-prone exam questions.

In total, the recommendation page was available to 1.08 million parents. Parents were prompted to open the page through in-App alerts and push messages. Our population consists of the 52K parents who entered the recommendation page in the period of data collection lasting from Jan 11th to Jan 29th, 2021.

The experimental arms are summarized in Table 1.1. The treatment assignment of each feature is independent of that of another feature. As such, the experiment is a fully-crossed design. Due to the experimental data yielding clear results for two features (recommendation reason, and the number of modules) in the first week, starting from Jan 23rd, these two features were set to values that optimized completion rates while other features' treatment assignment probabilities remained

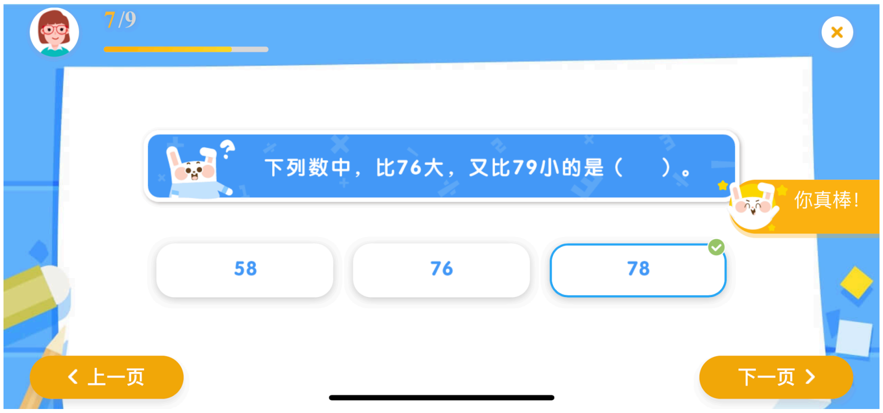

下列数中，比76大，又比79小的是（　）。

Figure 1.6: Question in an LPE module

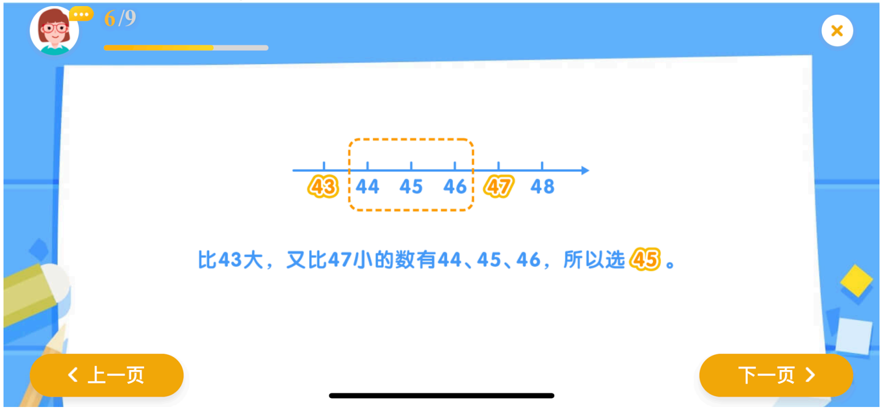

比43大，又比47小的数有44、45、46，所以选45。

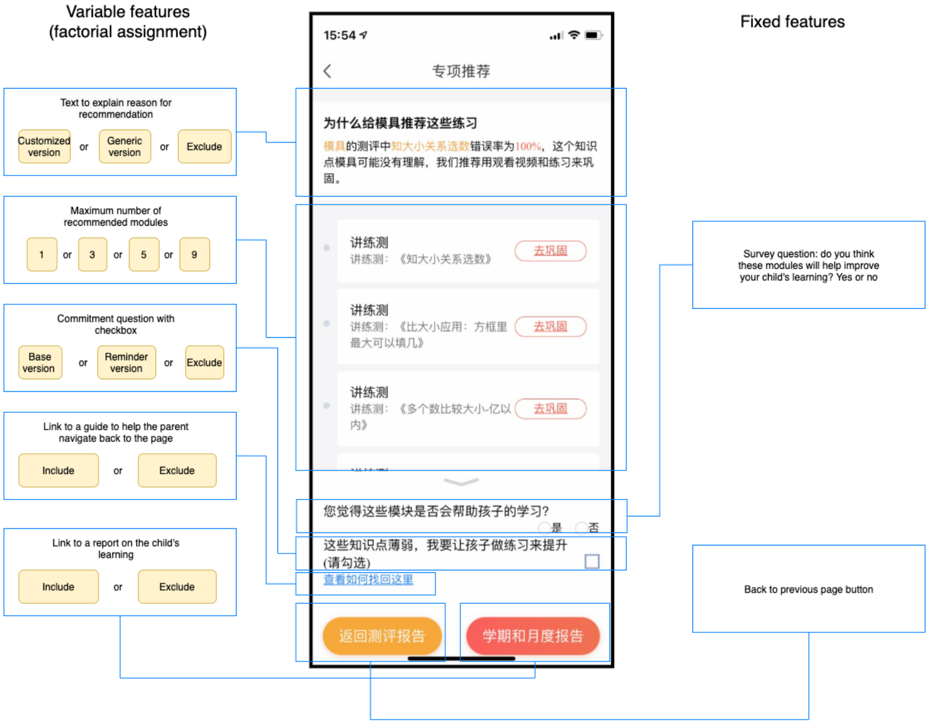
Figure 1.7: Explanatory text/audio in an LPE module

unchanged. Due to the need to experiment with various wordings of the commit question, the commit question feature was only finalized starting Jan 23rd. Hence, unlike the other features, the analysis of the commitment question feature uses only data from the 23rd onward.

Assignment value	Probability of assignment	Intervention
{Excluded, Customized version, Generic version}	$\{0.52, 0.24, 0.24\}$	Text to explain reason for recommendation
{1, 3, 5, 9}	$\{0.25, 0.25, 0.25, 0.25\}$	Maximum number of recommended modules
{Excluded, Base version, Reminder version}	$\{0.3, 0.35, 0.35\}$	Commitment question with checkbox
{Excluded, Included}	$\{0.5, 0.5\}$	Link to a guide to help the parent navigate back to the page
{Excluded, Included}	$\{0.5, 0.5\}$	Link to a report on the child's learning

Table 1.1: Recommendation page features experiment: Treatment assignment

Figure 1.8 shows how the different features in the way they appear in the parent App. On the left of the figure, we have mapped treatment conditions to sections of the screen they affect. On the right, we document the fixed features of the page that are shown to all users regardless of their treatment assignment.


Variable features
(factorial assignment)
Text to explain reason for recommendation
Customized version
or
Generic version
or
Exclude
Maximum number of recommended modules
1
or
3
or
5
or
9
Commitment question with checkbox
Base version
or
Reminder version
or
Exclude
Link to a guide to help the parent navigate back to the page
Include
or
Exclude
Link to a report on the child's learning
Include
or
Exclude
15:54
专项推荐
去巩固
去巩固
去巩固
是
否
返回测评报告
学期和月度报告
Fixed features
Survey question: do you think these modules will help improve your child's learning? Yes or no
Back to previous page button

为什么给模具推荐这些练习

模具的测评中知大小关系选数错误率为100%，这个知识点模具可能没有理解，我们推荐用观看视频和练习来巩固。

讲练测
讲练测：《知大小关系选数》

讲练测
讲练测：《比大小应用：方框里最大可以填几》

讲练测
讲练测：《多个数比较大小-亿以内》

您觉得这些模块是否会帮助孩子的学习？ ○是 ○否

这些知识点薄弱，我要让孩子做练习来提升(请勾选) □

查看如何找回这里

Figure 1.8: Recommendation page layout

Table 1.2 describes the outcomes and covariates used in this analysis. We use the extrapolated completion count as our primary outcome in this paper, which likely captures completion rates more accurately than the other measures. Nevertheless, when we use actual completion count and start counts, our results mirror the current results.

We do not have access to an indicator for whether the parent or the student actually completed the LPE modules. However, an LPE module takes more than two minutes to complete on average. Hence, although it is possible that parents were completing these modules, the vast majority of the completes were likely by students, not their parents. Nevertheless, the LPE module started metric is likely to have captured parents ”shopping around” by clicking on the modules to reveal their contents.

Variable type	Name	Description
Outcome	useful	Whether a user answered ”Yes” to the survey question (vs answering ”No”) conditional on responding.
Outcome	respond_survey	Whether a user responded to the survey question.
Outcome	n_complete_extrp	Extrapolated number of completed LPE modules. Since not all users who complete a course click on the ”complete” button. As such, this measure not only counts the ”complete” button events but also includes any start within 10 minutes of clicking a complete button.
Outcome	any_complete	Whether the user has completed any LPE modules.
Secondary outcome	n_complete	Number of completed LPE modules by counting of click on the ”complete” button at the end of each module. This serves as a robustness check for n_complete_extrp.
Secondary outcome	n_started	Number of started LPE modules, this measure will capture parents' ”shopping” behavior. This serves as a robustness check for n_complete_extrp.
Covariate	is_non_metro	Family lives in a non-metro county in China, typically an indicator for lower afluence.
Covariate	low_achievement	Child belongs to the bottom 30% of homework score distribution (on past exams in the current semester).
Covariate	high_achievement	Child belongs to the top 30% of homework score distribution (on past exams in the current semester).
Covariate	high_activity	Parents belonging to top 30% in number of days with any parent App activity or the parent has opened the learning report within the past two months.

Table 1.2: Recommendation page features experiment: Outcomes and covariates

Every feature in this experiment was designed to help improve parental satisfaction and follow through. Variations in the maximum number of modules displayed allow us to investigate the optimal number of recommended modules. Figure 1.9 explains the hypothesized causal pathways for each feature in the experiment.

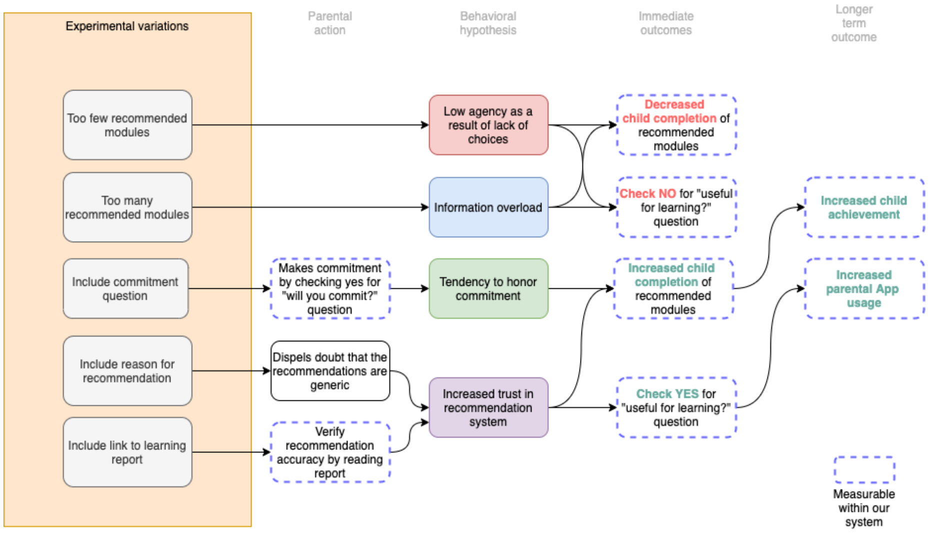

Experimental variations
Parental action
Behavioral hypothesis
Immediate outcomes
Longer term outcome
Too few recommended modules
Too many recommended modules
Include commitment question
Include reason for recommendation
Include link to learning report
Makes commitment by checking yes for "will you commit?" question
Dispels doubt that the recommendations are generic
Verify recommendation accuracy by reading report
Low agency as a result of lack of choices
Information overload
Tendency to honor commitment
Increased trust in recommendation system
Decreased child completion of recommended modules
Check NO for "useful for learning?" question
Increased child completion of recommended modules
Check YES for "useful for learning?" question
Increased child achievement
Increased parental App usage
Measurable within our system

Figure 1.9: Hypotheses for the function of recommendation page features

1.5 Results

In this section, we document the results of our experiments. Throughout this paper, error bars in figures correspond to 95% confidence intervals unless otherwise noted. Regression tables show standard errors in parentheses.

My research question is how do features on a page recommending custom exercise modules (LPE) affect parents' assessment of the usefulness of the recommended items and completion rates? The features we evaluate include:

1. Existence of text to explain the reason for recommending the modules.
2. Existence of a checkbox with a piece of commitment text.
3. The maximum number of recommended modules the page displays.
4. Existence of a link to a report on the child's learning.
5. Existence of a link to a guide to help the parent navigate back to the page.

Table 1.3 summarizes the main metrics for this experiment. We see that the response rate to the survey question is generally low, at 7% of all those who entered the page. But out of those responding, the vast majority 93% answer "YES" to the usefulness question. A little more than 4% of students actually complete any LPE module after their parents entered the page. We also note that there are more starts of LPE modules than completion, indicating potential "shopping" behavior on the part of the parents.

Event	N	Proportion	Notes
Page entry (unique parent)	51860	1.00	
Unique parent answer useful question	3600	0.07	
Unique parent answer "YES" to useful question	3342	0.93	Within parents who responded to survey question.
Unique parent check commit question	1087	0.02	
Unique student that completed any LPE module	2320	0.04	
N clicks on learning report	3043	0.06	
N clicks on how-to-find-back link	535	0.01	
N starts for LPE module	13739	0.26	
N completes for LPE module	7799	0.15	

Table 1.3: Recommendation page features experiment: Summary table

We document the sample sizes for all the treatment conditions of this experiment in Table 1.4. As all the treatment conditions were assigned orthogonal to each other, also known as a factorial design, a parent may constitute a unit in multiple treatment arms.

Feature	Treatment arm	Sample size
Recommendation reason	Exclude, Generic reason, Customized reason	13,280, 5,775, 5,955
Maximum number of modules	1, 3, 5, 9	6,238, 6,315, 6,306, 6,151
Guide to navigate back	Exclude, Include	25,945, 25,915
Link to learning report	Exclude, Include	26,058, 25,802
Commitment question	Exclude, Question without reminder, Question with reminder	5,426, 6,498, 6,510

Table 1.4: Recommendation page features experiment: Sample sizes

1.5.1 Main effects

We find a strong negative effect (70% drop) of providing a reason for recommendation at the top of the page on the number of LPE modules completed. We document this in Figure 1.10[1]. Why might this be the case? Consistent with the interpretation that adding these features induces cognitive overload, users who were shown a reason were also 60% less likely to answer the survey question (see Figure 1.11). Nevertheless, those who did answer provided statistically indistinguishable answers from the group not shown the reason (see Figure 1.12).

In Figure 1.10, we can also see that we find null effects on completion from adding 1) a commitment question, 2) a guide to navigate back to the page, and 3) a link to the child's learning report. Using our estimated standard errors, we were able to rule out effect sizes larger than 0.016 (±13%), 0.016 (±13%), and 0.032 (±18%) for these treatment conditions respectively.

Also consistent with the cognitive overload interpretation, we see significantly lower rates in answering the survey question from adding these features (see Figure 1.11). Similar to the previous result, we find statistically indistinguishable answers from the control group for all of these treatments (see Figure 1.12). As such, our hypothesized positive effects from these features are likely muted or even negated by the penalties they induced through cognitive load.

Unlike the above-mentioned features, we find no evidence that increasing the number of modules

[1] Effect sizes and standard errors estimates as OLS coefficient on the treatment variable (with feature = 1, no feature = 0) while including variables in the OLS regression for the other treatments in the factorial design.

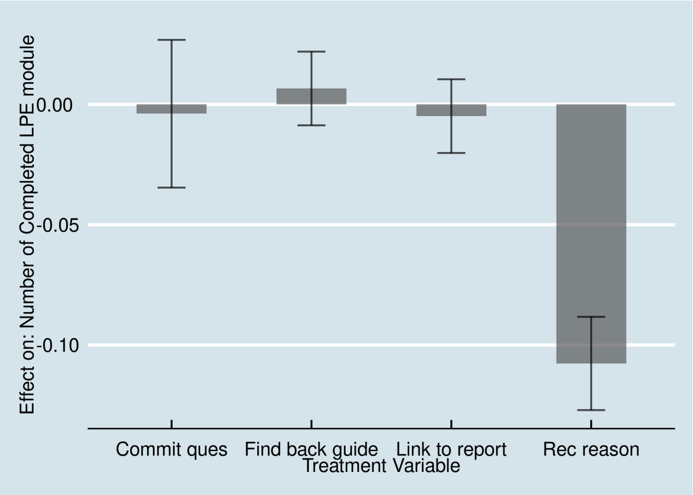


Figure 1.10: Effect of features on LPE module completion

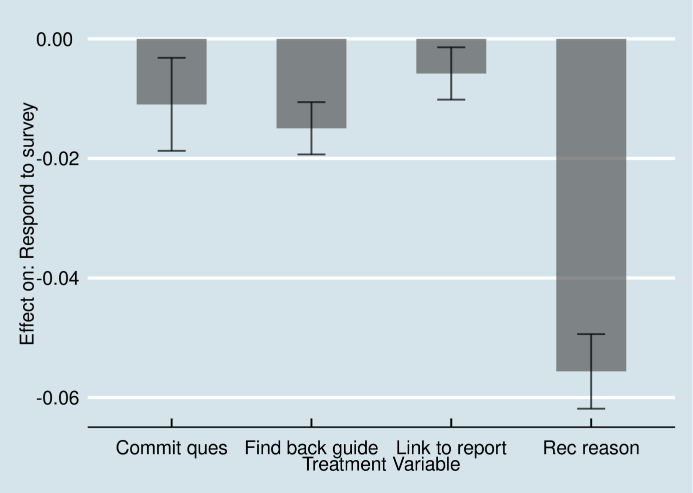


Figure 1.11: Effect of features on respond to survey

causes cognitive overload as they have statistically indistinguishable survey response rates (Figure 1.13) and affirmative answer rates (Figure 1.14)[2]. This suggests that increasing the number of homogeneous items such as LPE modules does not induce cognitive overload in the same way that the addition of other features does. It may take more cognitive capacity to read and understand features in different places on a page than to digest a greater number of similar items.

[2] Means are computed as the OLS coefficient on the binary indicators for the number of modules (1, 3, 5, 9) without intercept while controlling for orthogonal assignments of other pieces of the factorial design.

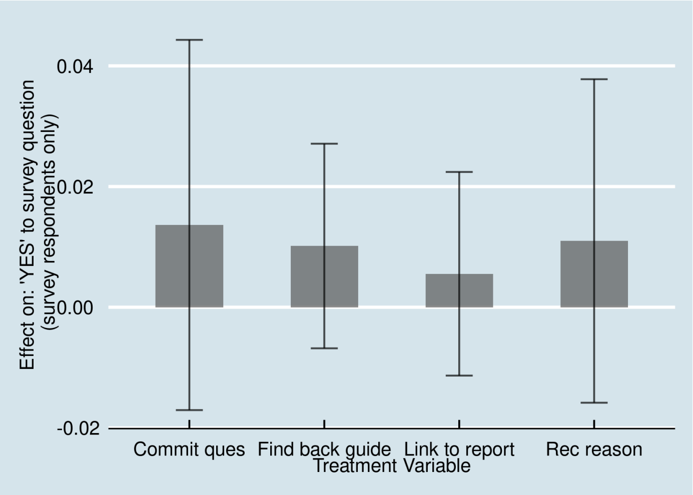


Figure 1.12: Effect of features on respond ”YES” to survey (vs respond ”No”)

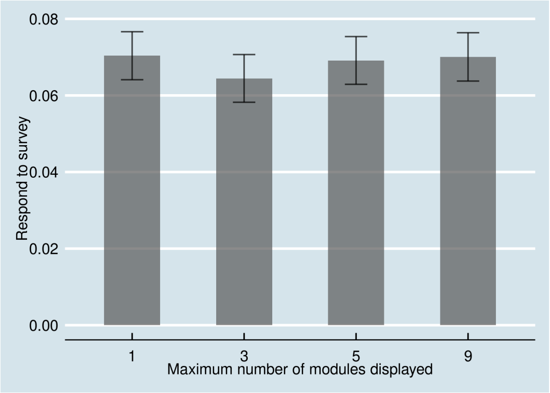


Figure 1.13: Respond to survey by number of modules

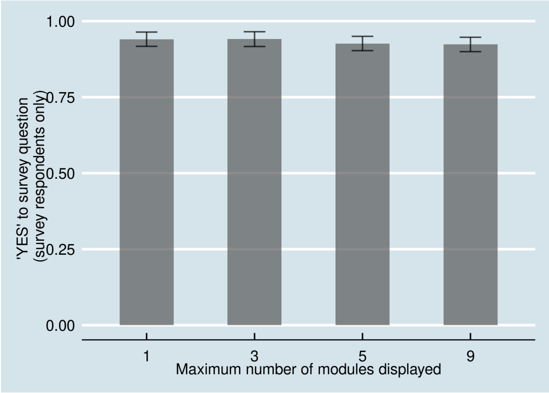


Figure 1.14: Respond ”YES” to survey by number of modules

While increasing the number of recommended modules did not increase the number of users completing any modules (Figure 1.15), but increased the number of completions by compliers (Figure 1.16). This effect is likely the mechanical consequence of giving users a larger set of choices.

1.5.2 Completion patterns by number of modules

Since we have shown that increasing the number of available modules does not increase the number of users completing any modules, it must have increased completion by increasing the number completed per user. Figure 1.17 shows how the distribution of completions per user in the different

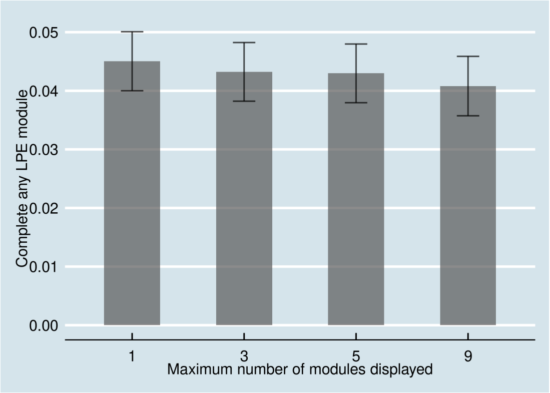


Figure 1.15: Completion of any LPE module by number of modules

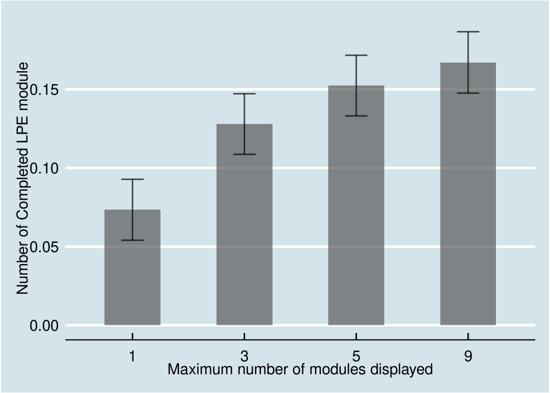


Figure 1.16: LPE module completion rate by number of modules

treatment groups.

These histograms reveal two prominent patterns. Firstly, the number of users declines almost monotonically as the number of completions increases. Secondly. there is a group of users who complete the maximum number of allotted modules as evident by the spike in the left-most bin of the histogram for users assigned to the 3, 5, and 9 groups. Taken together, these patterns suggest that the distribution of user preferences seems to be continuous and decreasing with the number of completions. Increasing the number of available modules increases completions by allowing those who want to complete more to do so.

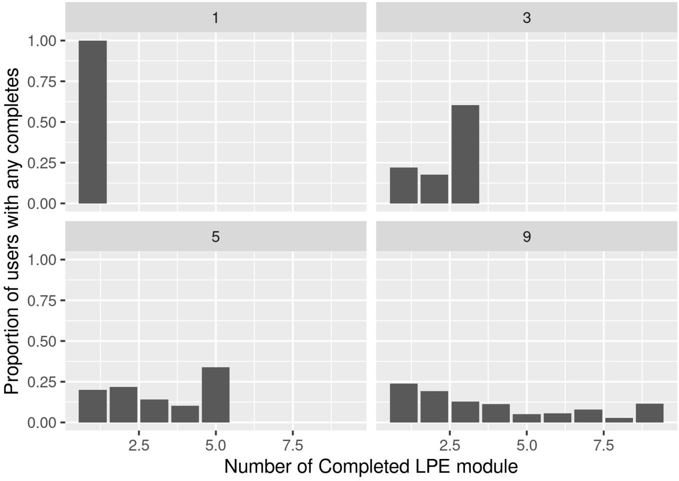


Figure 1.17: Number of modules completed by maximum number of modules available

1.5.3 Variations of features

In our experiment, the recommendation reason had two versions. The generic recommendation reason versions simply stated that the recommended LPE modules were based on the child's exam performance. The customized recommendation reason named the knowledge point that the student performed worse in and showed the percentage of questions within the knowledge point that the child had answered wrong. We see from Figure 1.18 that the effect on completion was statistically indistinguishable between the two versions of the recommendation reason.

The commitment question also had two variations. A base version asked the parent to commit to getting her child to complete the LPE modules. A reminder version also told the parent that she would be reminded. A reminder push message would be automatically sent to any parent who received the reminder version and checked the check-box. The reminder push message would be sent out at 8:30 PM on the same day the parent checked the check-box. We see from Figure 1.19 that the effect on completion was also indistinguishable between the two versions of the commitment question.

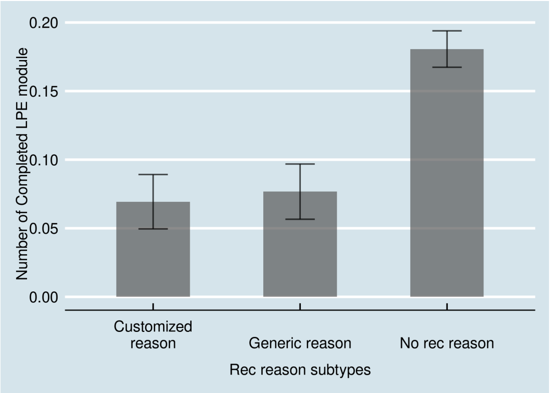


Figure 1.18: LPE module completion rate by recommendation reason type

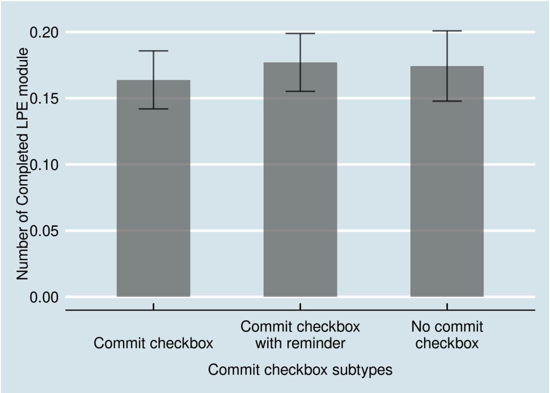


Figure 1.19: LPE module completion rate by commitment question type

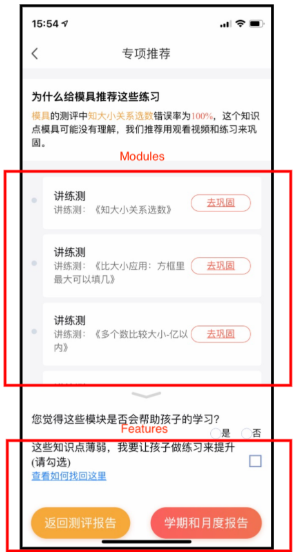
1.5.4 Modules vs features

We dig deeper into the differential effect on cognitive overload caused by two types of information present on the page. Having laid out the evidence in favor of a cognitive overload interpretation to our results, we seek to further examine how different types of information lead to cognitive overload.

Figure 1.20, using red boxes, highlights the two sections of the page in question. The first is the number of modules present in the middle of the page, the second is the section at the bottom of the page containing various features.

In this exercise, we want to examine how the survey response rate differs by the number of modules displayed and by the number of features available. The number of features is a count of the features available in the bottom section of the page for each user. We count the commitment question, the find back guide, and the link to the learning report. If all three features are present, then the count is 3; if only two out of the three are present, then the count is 2, and so on. Users who were assigned none of these features had a count of 0.

Figure 1.21 and Figure 1.22 shows the effects of the two types of information side by side. While increasing the number of modules shown did not decrease the survey response rate, increasing the number of features did. Low survey response rate can be a proxy for cognitive overload. As such, these results show that modules and features have different effects on cognitive overload. Perhaps because modules are homogeneous, increases in modules seem to require less cognitive energy to process than the increases in features.



为什么给模具推荐这些练习

模具的测评中知大小关系选数错误率为100%，这个知识点模具可能没有理解，我们推荐用观看视频和练习来巩固。

讲练测
讲练测：《知大小关系选数》

讲练测
讲练测：《比大小应用：方框里最大可以填几》

讲练测
讲练测：《多个数比较大小-亿以内》

您觉得这些模块是否会帮助孩子的学习？ ○是 ○否

这些知识点薄弱，我要让孩子做练习来提升(请勾选) □

查看如何找回这里

Figure 1.20: Features and modules

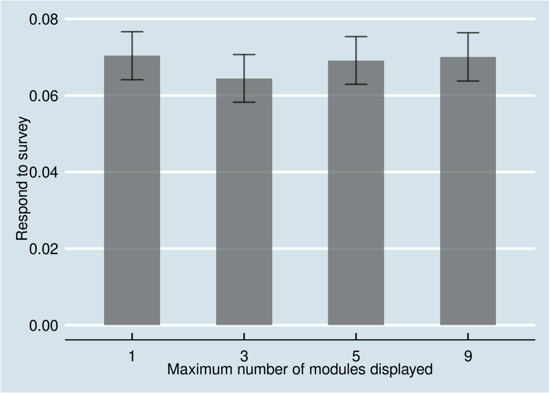


Figure 1.21: Respond to survey by number of modules

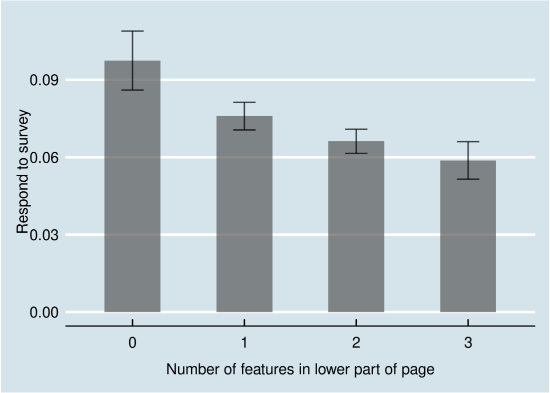


Figure 1.22: Respond to survey by number of features

1.6 Conclusion

To improve parenting through mobile apps, the parent needs to follow through and take action with their child to generate any meaningful impact on her. Our findings advance the literature on engaging parents through mobile devices. Our major contribution is demonstrating empirically that cognitive overload plays an important role in determining how much parents follow through. We also reveal that homogeneous modules do not incur cognitive overload in the same way that additional features do. As such, providing more homogeneous choices allow those who prefer to complete more to do so without introducing additional penalties on the rest of the users.

We showed that while many features may have hypothesized positive effects on parent follow through, the cognitive overload is an ever-present headwind. The hypothesized positive effects from any feature shown to the parent can be muted, or even negated, by the penalties they induce as cognitive load. Our findings should alert researchers and practitioners to think hard about the hypothesized benefits of a feature and weigh them against their potential cognitive load penalties. Above all, our results point to the importance of experimentation as it is almost impossible to establish the cognitive penalties of any feature a priori.

We also found that increasing the number of choices for homogeneous modules did not increase cognitive overload. But it increased follow through due to more choices being available. As we would expect with actions involving a cost in effort, the distribution of user preferences seems to be continuous and decreasing with the number of completions. Increasing the number of available modules increased completions by allowing those who want to complete more to do so. The implication for researchers and practitioners is that increasing the size of the choice set could induce more of the desired behavior. In our case, we did not find indications that doing so with homogeneous modules induced cognitive overload. However, our experiment capped the maximum number of modules shown at 9. These results may not hold in a setting where the choices are much more numerous than this.

Our results also revealed the modules and features seem to have a different effect on parents. Increasing the number of modules shown did not decrease the survey response rate but increasing the number of features did. Features seem to be more distracting than modules even though they occupy a smaller portion of the screen. This suggests that not everything on the screen is distracting. In our case, features may be more distracting than modules because parents are not familiar with their functionalities and need to devote more effort to understand them. Researchers and practitioners should be aware of these differences and critically evaluate features based on the mental effort needed to understand them.

Chapter 2

The Effect of Faculty Research on Student Learning in College

...

Chapter 3

Modeling Student Learning in Large-Scale Online Settings

3.1 Introduction

Online education has been transformed in recent years by the rise of mobile learning. Around the world, popular mobile learning apps offer students personalized paths of engagement. In India, the tutoring app Byju's personalizes students' learning journeys using a large knowledge graph (Bhatia, 2017). The US-based Khan Academy offers teachers the option of personalizing student assignments (Khan Academy, 2020b) and also allows students to choose their own pathway through practice exercises (Khan Academy, 2020c). Similar to Khan Academy, 17Zuoye offers choices to teachers and students in China (Sunny Education Inc., 2018). Together, these three apps account for more than 180 million users (Singh, 2020; Khan Academy, 2020a; Sunny Education Inc., 2018), and many other apps offer similar features to their users.

Choice over one's learning path is heralded as an essential part of a broader effort to improve learning through personalization. While this movement towards personalized learning is received with great fanfare in the online education industry, they also created important challenges for documenting student progress. Compared to the traditional setting of standardized tests, students are not exposed to the same items in personalized learning paths. Typically, students are only exposed to a small fraction of items from a large question bank. Furthermore, new interaction data is generated in real-time and new students and new items are frequently added to the system.

As an example, in 17Zuoye's database, there are 1.5 million unique questions exposed to students in December of 2017. However, among active users, the median student is only exposed to 398 questions over the same month. The high-frequency user at 95 percentile only logs 1115 questions in the same period.

Traditionally, education measurement tools based on Item response theory (IRT) are designed for standardized tests with dense data. They are unable to scale to this setting where the student-item matrix is large and sparse. The literature on linking has procedures for accommodating data sparsity. But existing methods are ad hoc, heavily dependent on model specification, and do not computationally scale to large data.

In this paper, we propose new models based on latent factorization and Bayesian variational inference to address these challenges. We find that our models retrieve true parameters with greater fidelity than traditional models in small data settings (section 3.5). In addition, our models also scale well computationally to industrial-size datasets (section 3.6). Compared to the two-parameter model, our factorization models are able to make more accurate predictions on the hold-out test set in general. More latent factors and hierarchical dependence on question attributes contribute to better predictive performance in lower-frequency content areas (section 3.7). Benchmarking against similar models from the recent literature, our models are much faster to run and produce better predictions (section 3.8).

In section 3.9 of our paper, we describe a real-world application of our models. *Smart Homework* uses predictions from our models to make personalized question recommendations to students that are not too hard nor too easy. We plan to experimentally test the effect of personalization on student outcomes in a future study. Section 3.10 concludes the paper by discussing other potential applications of our models.

3.2 Related Work

Our methods are inspired by recent advances in inference techniques and machine learning models that employ user/item factorization. One central idea behind our proposed models is that students and items have latent vector representations whose dot product influences the probability of answering a question correctly. This approach draws from recent work on latent factorization that have been applied to providing online recommendations (Gopalan et al., 2015; Donnelly et al., 2020), analyzing complementarity and substitutability in consumer choice (Ruiz et al., 2017; Donnelly et al., 2019), and geographical preferences of restaurant-goers (Athey et al., 2018).

These works draw from the large literature on recommender systems where the standard approach is to find a try to find an approximation of the full matrix of user-item interactions using the product of two lower-rank matrices. Nonetheless, these recent works extend the latent vector representation approach to allow latents to depend on observed characteristics and to account for time-varying effects. We will also incorporate these innovations into our approach.

Advances in Bayesian variational inference make Bayesian inference computationally feasible on massive datasets. Variational inference recasts Bayesian inference as an optimization problem, lending it to stochastic optimization techniques (e.g. stochastic gradient descent) which allows the algorithm to scale to large datasets. See Blei et al. (2017) for a review. Recent engineering work allows models using stochastic variational inference to be implemented through off-the-shelf machine learning packages (Tran et al., 2016; Bingham et al., 2019).

Our work is also related to the large literature on linking in educational measurement. Linking refers to the practice that compares student performance across different tests (see Kolen and Brennan (2004) for a review). Similar to our goal, linking can be interpreted as a way to overcome sparsity in the combined student-item matrix of different tests that share common persons or common items (Reardon et al., 2019). A major difference, however, is that the linking literature is generally focused on the design of tests for particular settings where researchers control the recruiting of examinees and the administration of exams. As such, procedures for accommodating data sparsity are mostly ad hoc (e.g. estimating parameters from test A first, then keeping these parameters fixed when estimating test B) Kolen and Brennan (2004). There is no prevailing consensus on which methods should be used and typical works in this literature are also only applicable to certain model specifications. Our work differs from this literature in the scale of data and the highly sparse nature of our context and the generality of our approach.

Our work is also related to the literature that casts the model of student learning as solutions to a predictive problem (see Pardos (2017) for a review). A prominent line of research in this realm centers around Bayesian Knowledge Tracing and builds temporal models of student learning (Corbett and Anderson, 1994). Recent advances along this line of research have employed deep and recurrent representation to this task (Piech et al., 2015), incorporating prior knowledge of the learners into the model (Yudelson et al., 2013), and modeling question difficulty (Pardos and Heffernan, 2011).

Many researchers used the temporality of student responses and leveraged recent developments in training deep neural networks to predict student-question responses (see Hernández-Blanco et al. (2019) for an extensive review). However, recent papers find that simpler models with psychological interpretations can behave just as well as deep learning approaches when structured to fit a few regularities (Khajah et al., 2016; Wilson et al., 2016a,b).

Our approach takes the middle ground where we produce interpretable parameters while using a factorization approach to flexibly model latent regularities. Our work essentially reduces the dimensionality of the temporal student-question sequence into a student-question matrix and attempt to solve a matrix completion problem (Candes and Plan, 2010; Mazumder et al., 2010). One of the major hurdles in recent work in this line of literature is that computational costs may be prohibitively high for large datasets using existing approaches (e.g. (Poole et al., 2008; Bailey, 2007; Shor and McCarty, 2011)). To remedy this, Imai et al. (2016) proposed a series of EM algorithms that significantly improved the computational performance in the estimation of ideal points (estimating latent traits for legislators and bills) by as much as 1000 times over existing MCMC approaches. Most recently, computer scientists Wu et al. (2020) proposed optimizing a specialized loss function that lower bounds the marginal likelihood over a student' s responses. This loss is named Variational approach for Item response theory based on a novel lower BOund (VIBO). The authors showed that VIBO scales well to predictive exercises in large real-world data such as the PISA science dataset.

3.3 Model specification

We use Bayesian variational inference for all parameters in our proposed models. See section 3.11.1 for an summary of Bayesian variational inference. We place Gaussian posteriors on every parameter with flexible mean and scale. We also initialize most parameters with Gaussian priors with mean 0 and a standard deviation of 1. In practice, our optimization methods also use stochastic gradient descent and various computational optimizations implemented by Bingham et al. (2019).

3.3.1 Model 1: Two-Parameter model

Our first model takes the form of the classic Two-Parameter model from the item response literature. The only difference to traditional models is that we estimate the parameters through Bayesian variational inference.

In this model θ_i is the student parameter and α_j, β_j are question parameters. For the probability of student i answering question j correctly ($Y_{ij} = 1$), the two-parameter model is defined as:

$$P(Y_{ij} = 1|\theta_i, \alpha_j, \beta_j) = \frac{1}{1 + \exp(-\alpha_j(\theta_i - \beta_j))}$$

3.3.2 Model 2: Latent factorization model

In this model, we map students and questions into latent vectors ($\boldsymbol{\theta}_i, \boldsymbol{\alpha}_j$) and allow their inner product to influence the probability of correctness. The model is given by:

$$P(Y_{ij} = 1|\boldsymbol{\theta}_i, \boldsymbol{\alpha}_j, \beta_j) = \frac{1}{1 + \exp(-(\boldsymbol{\theta}_i^\top \boldsymbol{\alpha}_j - \beta_j))}$$

The main benefit of latent factorization in this model is analogous to that of methods commonly found in recommendation systems. These types of models allow the parameters to learn from the structure of the student-question matrix. The richness of the latent factors allows us to quantify student ability in multi-dimensional ways as the predictions for correctness will be different for items with different latent factors.

3.3.3 Model 3: Hierarchical factorization model

In this model, we retain the structure of the latent factorization model. However, we replace a simple latent vector $\boldsymbol{\alpha}_j$ with a concatenation of the latent $\boldsymbol{\alpha}_j$ and a latent transformation of observed question covariates X_j through the transformation matrix H_α. We denote this concatenated vector as $\boldsymbol{\alpha}_j \oplus H_\alpha X_j$. Accordingly, we make $\boldsymbol{\theta}_i$ to be the same length as the result of the concatenation.

The trainable parameters of this model are $\boldsymbol{\theta}_i, \boldsymbol{\alpha}_j, H_\alpha, \beta_j$. The model is given by:

$$P(Y_{ij} = 1 | \boldsymbol{\theta}_i, \boldsymbol{\alpha}_j, H_\alpha, \beta_j, X_j) = \frac{1}{1 + \exp(-(\boldsymbol{\theta}_i^\top (\boldsymbol{\alpha}_j \oplus H_\alpha X_j) - \beta_j))}$$

By adding a flexible dependency on the observed characteristics of questions, we are allowing question attributes to influence the probability of correctness directly. Similar to Athey et al. (2018), this hierarchical structure may allow the model to perform better, especially for questions that appear in the data with low frequency.

3.3.4 Software implementation

Our code base[1] uses the probabilistic programming package Pyro (Bingham et al., 2019) for Bayesian stochastic variational inference. Pyro is built on top of Pytorch (Paszke et al., 2017) and uses data structure, automatic differentiation, and optimizers from the latter.

3.4 Data collection

We apply our models to data generated on the 17Zuoye platform. The data comes from homework assignments and exams in three subject areas–English, math, and Chinese. Records are logged at the students-question level.

Exam data on the 17Zuoye platform are also tagged with question attributes. Attributes include the appropriate grade level of the question. They also include two types of domain knowledge tags. One system maps questions to competencies. The other maps questions to skills. These tags are manually labeled by content specialists.

[1] Code access is available at `https://github.com/henrishi/bm_model`.

3.5 Parameter retrieval

In this section, our goal is to compare how well our proposed models and estimation strategies retrieve true data generating parameters. In addition, we also compare the parameters retrieved by our approach to those retrieved by a widely-used traditional item response model package *ltm* (Rizopoulos, 2006). *ltm* results are labeled as *traditional_2param* in the following figures.

We focus on the two-parameter models for this exercise because the factorization models are under determined systems and there are multiple parameter arrangements that can yield the same prediction. *ltm* produces frequentist point estimates and standard errors. For the sake of comparison, with our Bayesian models, we take the mean of the posterior distribution as our estimator and the standard deviation of the posterior distribution as our standard error.

3.5.1 Simulated dense data

We fit *ltm* and *Bayesian two-parameter model* on a small simulated data set. Our data generating process samples θ and β from a normal distribution with mean 0 and standard deviation of 1. α is sampled from a normal distribution with mean 1.2 and standard deviation of 0.5 while constrained to be positive. The student-question response data is then drawn from a Bernoulli distribution where the probability of for a correct answer from student i and question j is $P(Y_{ij} = 1) = \frac{1}{1+\exp(-\alpha_j(\theta_i - \beta_j))}$. We have 30 questions and 50 students in the simulated data and every student has a response for every question.

We focus on the estimate for the β parameter since it is the easiest among the three parameters to estimate. Figure 3.1 compare the estimates from *ltm* and *Bayesian two-parameter model* alongside the true data-generating parameter. Table 3.1 presents the correlation (both linear and ranking correlations) between the estimates and the true parameter values.

We see that the *Bayesian two-parameter model* is recovering parameters better than *ltm* . *ltm* has large devious estimates for the parameter with the lowest value and large error ranges for certain parameters with mid-range values. Not only are the point estimates more devious than *Bayesian two-parameter model* , but the standard errors are also bigger for *ltm* in general. This is especially pronounced for the devious estimates. These deviations hurt the correlations of the estimates from *ltm* with the true parameter across all the correlation metrics we report (Pearson, Kendall, and

Spearman).

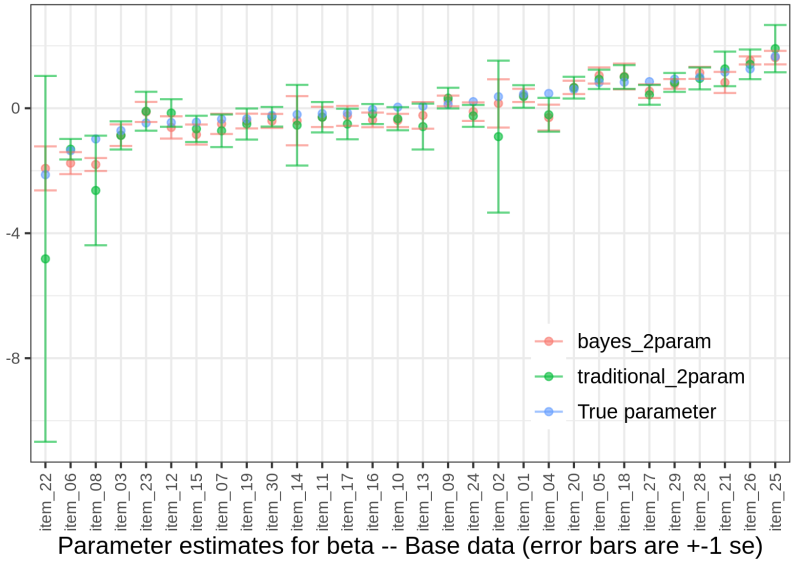


Figure 3.1: Small dense data: true and estimated beta

Stats	Pearson correlation (linear)	Kendall correlation (ranking)	Spearman correlation (ranking)
Bayesian two-parameter model - True parameters	0.95	0.83	0.93
ltm - True parameters	0.91	0.69	0.83

Table 3.1: Small dense data: correlation stats for beta

3.5.2 Simulated overlap data with missings

In the current exercise, we generate a simulated dataset where two groups of students share some overlapping questions but not others. The way missing data is structured is shown in Figure 3.2. Our data generating process samples θ from a normal distribution with mean 0 and standard deviation of 1. α is sampled from a normal distribution with mean 1.2 and a standard deviation of 0.5 while constrained to be positive.

Different from the section 3.5.1, we sample the β for overlap questions and those only available for students 1 - 50 from a normal distribution with mean 0 and standard deviation of 1. However, we sample the β for questions only available for students 51 - 100 from a normal distribution with mean 0.5 and standard deviation of 0.7. We sampled the β parameters this way to mimic real-world settings where question difficulty is usually different for different groups of students. As in 3.5.1 the student-question response data is then drawn from a Bernoulli distribution where the probability of a correct answer is a logistic function of the parameters.

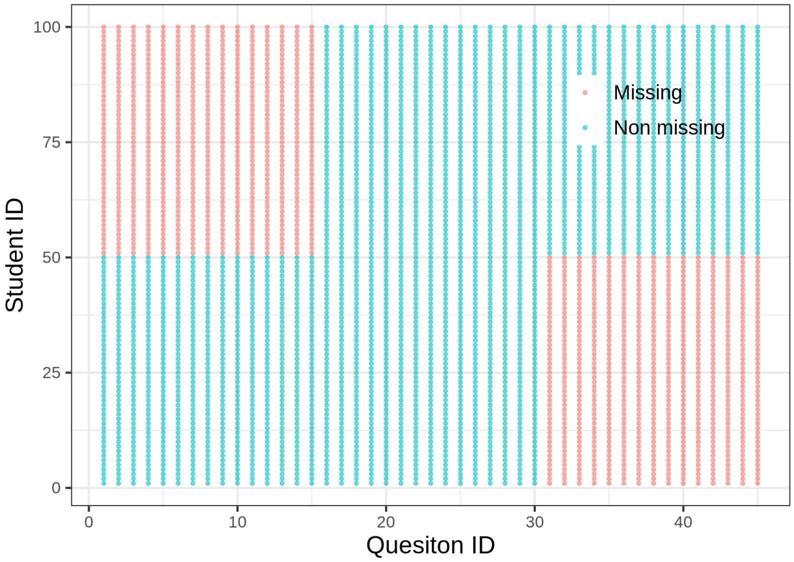


Figure 3.2: Overlap data missing pattern

In this setting, we again compare the performance of the *ltm* model to the *Bayesian two-parameter model* model in estimates of β. We find that the problem of devious estimates is greatly exacerbated for *ltm* . Figure 3.3 shows that the error ranges are excessively large for some estimates from *ltm* .

For a more informative figure, we take out the standard error bars of the outliers from the *ltm* model and re-plot in Figure 3.4. We see that while the majority of parameters have similar under both models, *ltm* is yielding very devious estimates for items 11, 31, and 37. These items are not overlapping items (see point map above) and are only taken by a single group of students.

Bayesian two-parameter model produces point estimates that are much closer to the true parameters for the parameters that *ltm* failed to estimate accurately. This corroborates the Bayesian inference property that the prior distribution serves as a regularizer and allows the model to be more numerically stable. Finally, we see that *Bayesian two-parameter model* again dominates *ltm* in correlation stats with the true parameters by a large margin in this data setting.

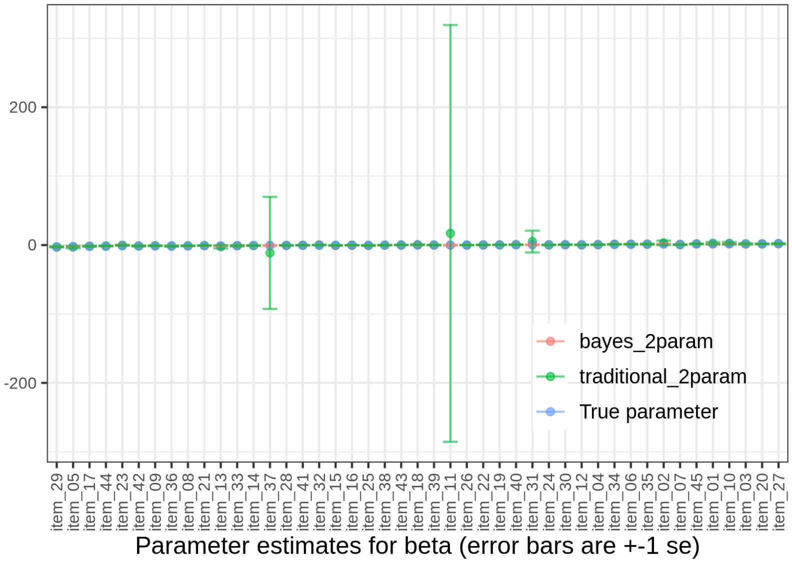


Figure 3.3: Overlap data: true and estimated beta

Stats	Pearson correlation (linear)	Kendall correlation (ranking)	Spearman correlation (ranking)
Bayesian two-parameter model - True parameters	0.96	0.84	0.96
ltm - True parameters	0.41	0.76	0.91

Table 3.2: Overlap data: correlation stats for beta

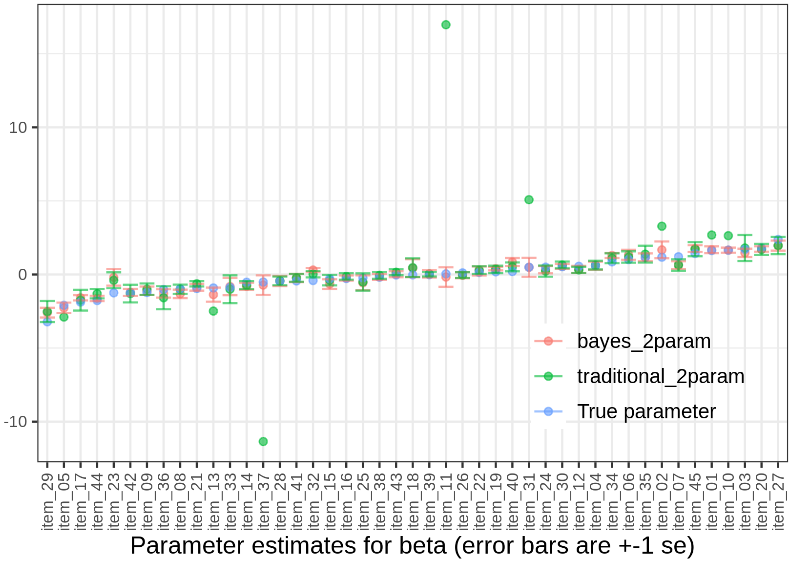


Figure 3.4: Overlap data: true and estimated beta (zoomed in)

3.6 Computation performance

In this section, we report the computation performance of our proposed models using an industrial-scale dataset with 88 million responses. The purpose of this section is to show that our proposed Bayesian models *Bayesian two-parameter model* and *Bayesian factorization model* are computationally tractable for real-world applications. Traditional methods such as *ltm* cannot be compared here because they could not handle data of this size. Specifically, *ltm* is unable to produce any useful estimates for even very small datasets when sparsity is at this level. [2]

This dataset comes from homework records of 1000 schools over a period of 2 months. In this dataset, we have 334K questions and 162K students. The overall density of the student-question matrix is 0.16%. The training of our models terminates when convergence has been achieved. We define convergence as the change in loss averaged over the last 5 iterations falls below 0.1% of the change in loss from the first to the second iterations.

In Figure 3.5 we show the training loss for different model specifications over time. Even though we adopt stochastic gradient descent for optimization, the overall loss curve is smooth due to the large amount of data used for training. The more complex models, factorization models with longer latent vectors, tend to take marginally longer to train. Most models converge within 30 minutes, the longest model to train took less than 40 minutes (see Figure 3.6).

3.7 Predictive performance

We compare the prediction performance of our proposed models using a large exam dataset where we have access to question attributes. We want to compare the performance of all three models, *Bayesian two-parameter model* , *Bayesian factorization model* , and *Bayesian hierarchical factorization model* . Since *Bayesian hierarchical factorization model* needs question attributes as inputs, we needed to test our predictive performance on a dataset that has question attributes. The exams

[2] *ltm* produces numerical errors for even very small samples from this dataset (e.g. a sample of 1000 records). For example, a random sample of 1000 records translated into a matrix of 639 students by 717 questions. *ltm* returns numerical errors for the resulting matrix. One can get *ltm* to run if the missing entries are replaced values (e.g. change all missing values to 0), but running *ltm* on the resulting data takes 16 minutes to converge. My testing shows that *ltm* convergence time is roughly $O(n^2)$ meaning that doubling the amount of data takes 4 times as long to converge. This is an exorbitant amount of time considering that 1000 records are a mere 0.00011% of the full dataset with 88 million records.

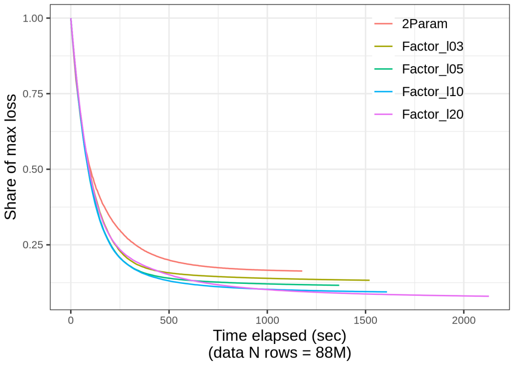


Figure 3.5: Training loss

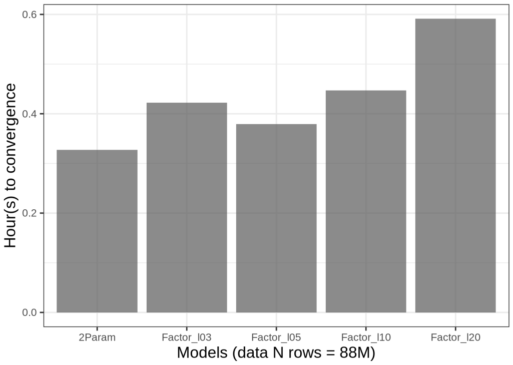


Figure 3.6: Convergence time

are also higher stakes than typical homework assignments, as such the data may be more reflective of actual competencies and less noisy as a result.

In this dataset, we have 8.6 million student-question records. There are 3.6k questions and 261k students in total. The overall density of the student-question matrix is 0.92%.

The data were randomly divided into 80% training, 10% validation, and 10% test sets at the level of student-question interactions. This means data for a single student may appear in any of the three sets. The same goes for data from a single question. Model training would stop automatically once convergence is achieved. To get a better metric of the models' capabilities, we define a stricter convergence criterion – as average per-iteration changes in loss becoming 0.05% the initial change.

We compare *Bayesian factorization model* against *Bayesian two-parameter model* in sections 3.7.1 and 3.7.2. Having established the superiority of factorization models, we move on to quantify the gains from adding hierarchical dependency in *Bayesian hierarchical factorization model* in section 3.7.3.

3.7.1 Overall predictive performance

We first document the overall predictive accuracy for *Bayesian two-parameter model* and *Bayesian factorization model* in Figures 3.7 and 3.8. For *Bayesian factorization model* , we show results from three models where the length of the latent vectors $\boldsymbol{\theta}, \boldsymbol{\alpha}$ are taken to be 3, 5, 10, and 20 respectively. We see that *Bayesian factorization model* models enjoy higher AUC in the test set than *Bayesian two-parameter model* . The F1 statistic tells the same story where *Bayesian factorization model* models are superior predictive performance.[3] We note that in all models, training set accuracy is higher than the test set. This is partially attributable to the sparse nature of the student-question matrix. Some students or questions may only show up in the training set or the test set, making test set predictions less accurate than the training set.

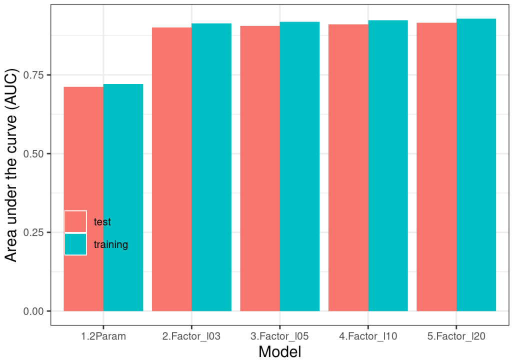


Figure 3.7: Area under the curve (AUC) by model

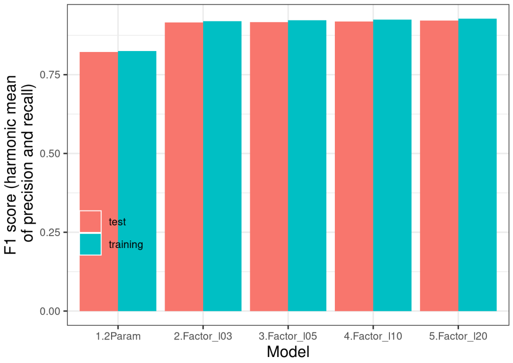


Figure 3.8: F1 statistic by model

3.7.2 Predictive performance by content area

Do factorization models with more latent factors perform better than simpler models on predictive accuracy in less frequent content areas. In this section, we answer this question by examining the predictive performance of candidate models by question knowledge labels. We have 22 knowledge labels in our data. For example, a knowledge label for a math question may be "arithmetic" or "geometry", one for an English question may be "English spelling" or "English pronunciation".

Figure 3.9 shows the distribution of student-question records by knowledge labels. From left to right, we see knowledge labels in descending popularity. We note that some knowledge labels have significantly lesser data than the most popular knowledge labels.

We document the predictive performance of different models across knowledge labels in 3.10 and 3.11. We see that confirm that *Bayesian factorization model* dominates *Bayesian two-parameter model* in both AUC and F1 across the knowledge labels. Interestingly, the more complicated factorization models (the 10-factor and 20-factor models) have better performance than simpler factorization models (the 3-factor and 5-factor models) in predictive performance for the less frequent

[3] In calculating the F1 statistic, we set the predictive threshold at $P > 0.5$ for a positive prediction.

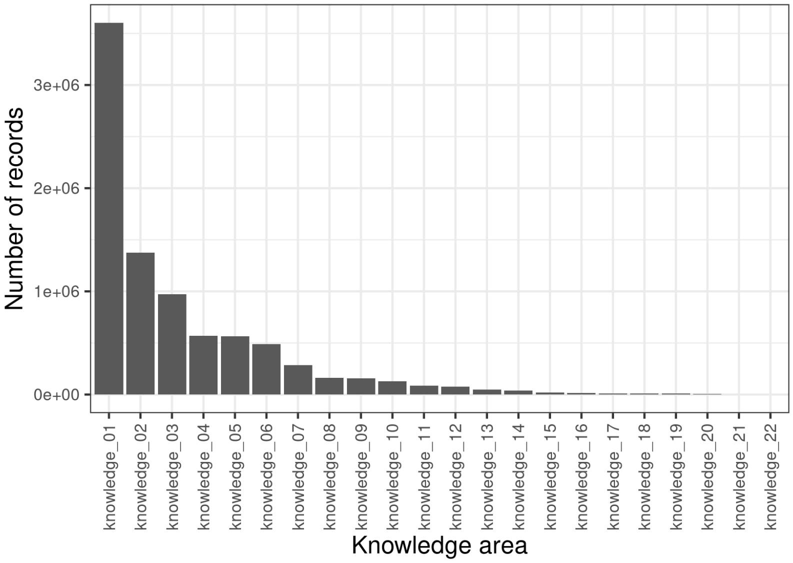


Figure 3.9: Count of student-question records by knowledge label

knowledge labels. This suggests that additional latent factors may have picked up additional heterogeneity useful in predicting less frequent knowledge labels.

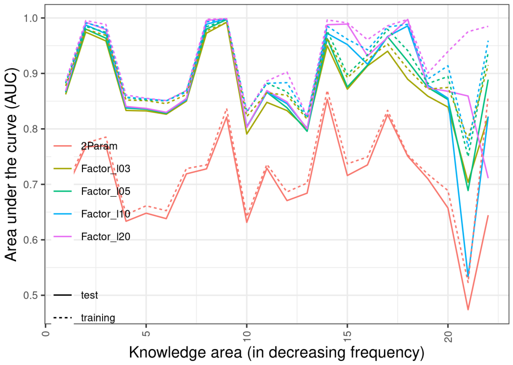


Figure 3.10: Area under the curve (AUC) by knowledge label

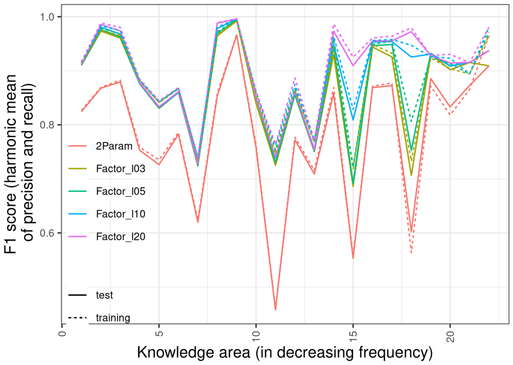


Figure 3.11: F1 statistic by knowledge label

3.7.3 Performance gains from adding hierarchical dependency

Having established the superiority of factorization models over *Bayesian two-parameter model* , we move on to quantify the gains from adding a hierarchical dependency in *Bayesian hierarchical* *factorization model* . To make models more comparable, we hold the size of latent vector constant and compare the *Bayesian factorization model* model with the same model that adds different hierarchical dependency.

We use two types of question attributes, one of which is the one-hot vector of knowledge label described in section 3.7.2. The other attribute is a multi-hot vector of skills involved in the question. The attribute is encoded as a multi-hot vector because a single question can be associated with multiple skills. For our candidate models with hierarchical dependency, we have freedom in choosing the size of the latent matrix H_α. Following Athey et al. (2018) we pick H_α such that some entries are 0 so that certain types of question attributes can only contribute to certain parts of the resulting latent vector $H_\alpha X_j$. As specified in 3.3.3, the latent vector $H_\alpha X_j$ is a linear combination of these representations. We choose two model specifications, the first structures H_α such that the knowledge labels and skills each map to one latent scalar so the resulting $H_\alpha X_j$ is a length 2 vector. The second is slightly more complex in that the knowledge labels and skills each map to two latent scalars so the resulting $H_\alpha X_j$ is a length 4 vector.

In Figure 3.12 we compare the factorization model with a length 3 latent vector to a hierarchical model with the same setup. We see that the hierarchical models outperform the factorization models slightly overall, but the improvement is more significant for certain low-frequency knowledge labels. The same is true when we look at factorization model with a length 5 latent vector and hierarchical models with the same set up (see Figure 3.13).[4]

[4] For the sake of brevity, we focus on AUC for our comparison, but the F1 statistics results are substantively the same as the AUC results.

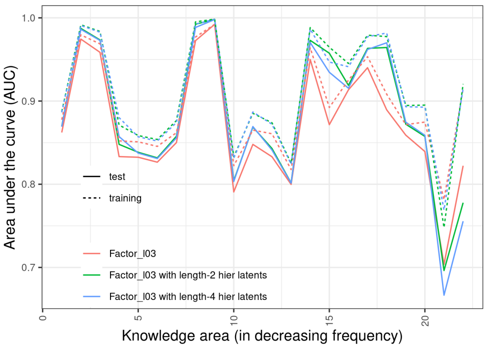


Figure 3.12: Area under the curve (AUC) by knowledge label (models where latent vector size = 3)

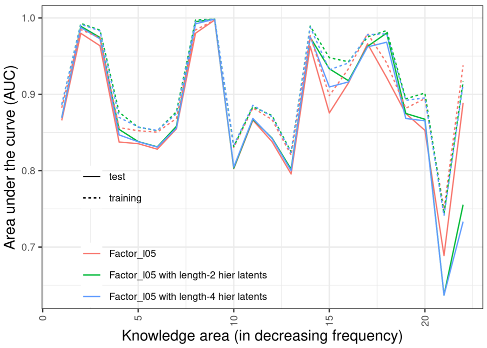


Figure 3.13: Area under the curve (AUC) by knowledge label (models where latent vector size = 5)

3.8 Benchmark comparisons

In this section, we compare our model against prominent contenders in recent literature on fast, accurate, and scalable IRT model estimations. Imai et al. (2016) proposed a series of EM algorithms that significantly improved the computational performance in the estimation of ideal points (estimating latent traits for legislators and bills) by as much as 1000 times over existing MCMC approaches. The work is highly cited in the political science literature (hereafter *emIRT*). Most recently, Wu et al. (2020) proposed coupling a specialized loss function named VIBO with variational inference to estimate IRT models (hereafter *varIRT*). They have shown that their model works well in parameter retrieval exercises with simulation data and scales well to predictive exercises with real-world data.

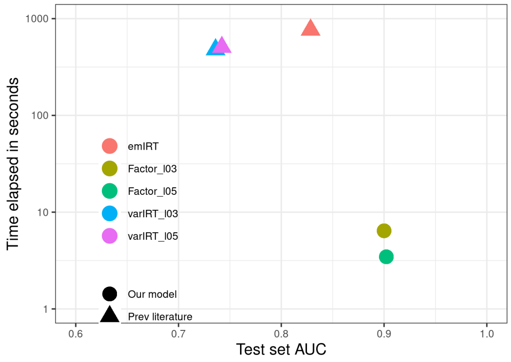


Figure 3.14: Computational and predictive performance: proposed models vs literature benchmarks

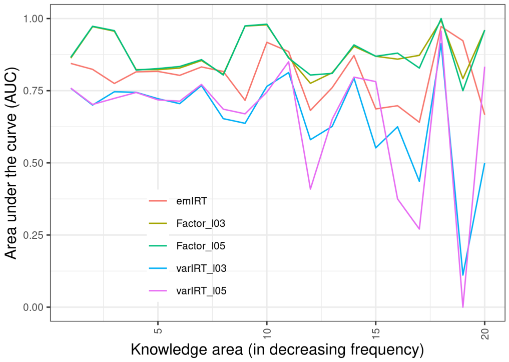


Figure 3.15: Test set AUC by knowledge label: proposed models vs literature benchmarks

In this exercise, we want to run both *emIRT* and *varIRT* on the same dataset with the same computer as we used for our previous predictive exercises (8.6M student-question records) and compare the models' computational and predictive performance. While we attempted to run all models on the whole dataset, *emIRT* and *varIRT* could not digest the entire dataset as it was too massive and demanded resources beyond the physical resources of the computer.[5]

As a result, we trained all our models using a smaller subset of the data. The subset was

[5] Both *emIRT* and *varIRT* resulted in out-of-memory errors on our computer, which had 16G of memory and was able to perform estimations with our proposed models with ease. One possible contributor to the disparity is that *emIRT* and *varIRT* both require data to be fed to it in wide form where each row represents a student and each column a question. This greatly increases the size of the data in memory compared to the long format feed data that our implementation adopts.

created by sampling 10K students from the original dataset at random and keeping all the questions associated with each student. The resulting data has $327,082$ student-question records, covering $10,000$ student and 3201 questions. The student-question matrix has a density of 1.02%. We randomly split the data into 90% training set and 10% test set.

Similar to our model, *varIRT* allows for more than one latent dimension. As such, we ran *Bayesian factorization model* and *varIRT* each with 3 and 5 latent dimensions. The official implementation of *emIRT* only allows for one latent dimension so we did not produce variations of the model. Note that we use *Bayesian factorization model* and not *Bayesian hierarchical factorization model* in this exercise because *varIRT* does not support hierarchical dependency on observables and using such would result in an unfair comparison.

Figure 3.14 shows how our model compares to *emIRT* and *varIRT* in both computational performance and test set predictive performance (AUC). We see that *Bayesian factorization model* dominates both alternatives with faster computational performance and better predictive accuracy. While the lead in predictive performance over *emIRT* and *varIRT* is considerable, both versions of the *Bayesian factorization model* also represent a substantial gain in computational performance. They ran on the data in under 10 seconds, while *varIRT* and *emIRT* took around 10 times the amount of time to run – on the order of hundreds of seconds (the y axis of Figure 3.14 is log scaled).

Figure 3.15 shows how our model compares to *emIRT* and *varIRT* by predictive performance across knowledge labels. Barring a few knowledge labels where *Bayesian factorization model* had similar performance with *emIRT* , the former largely dominated both *emIRT* and *varIRT* . Notably, despite being more complex in model specification, *varIRT* underperformed *emIRT* in high-frequency knowledge labels and most of the low-frequency knowledge labels.

3.9 Model application: recommender system for personalized questions

This section describes a practical application of our proposed models. Our models were put into use at our partner company to power a recommender system producing personalized questions for students. The recommender system is one piece of a broader product, *Smart Homework* , which combines personalized questions with immediate feedback when the student incorrectly answers a question. Figure 3.16 shows the relationship between the different components of the system.

The recommender system aims to produce questions that are not too difficult nor too easy. This aligns with a long line of research in educational psychology stemming from the Zone of Proximal Development proposed by Vygotsky (1980). The predicted probabilities for student i getting question j correct is computed using our proposed models from historical data.

Once the probabilities are calculated, the question pool is filtered by taking out questions that the student has already attempted in the past. Then the pool goes through a ranking algorithm for each student. The algorithm ranking questions based on a weighted sum of different factors. The most prominent factor is how close the predicted correctness probability is to 0.7. Other factors include how recent was the question created, whether the question includes a picture or a table. The top 10 questions are selected to form the *Smart Homework* .

As the student works through her *Smart Homework* she moves from one question to the next if she answers correctly. When she answers a question incorrectly, she is taken through a series of explanations and exercises also known as the learn-practice-explain (LPE) module. The student is guided through a series of questions and animations that explains one knowledge point for the triggering question (see Figure 3.17). The typical time to complete this module is between 2 and 4 minutes. As the student answers the questions, only a correct answer would allow the student to move forward. A wrong answer would trigger a hint and the student is directed to answer the question again.

After completing the module, the student is asked to answer another question similar to the original question she missed. Regardless of whether she answered this question correctly, she is taken back to other questions in the *Smart Homework* . *Smart Homework* exits when the student has completed every question.

3.9.1 Batch updates and parameter freeze

We added two features to the implementation of our models to address practical challenges for building *Smart Homework* . Firstly, we implemented batch updating of parameters. Since our models are based on Bayesian inference, making updates to parameter estimates is very straightforward.

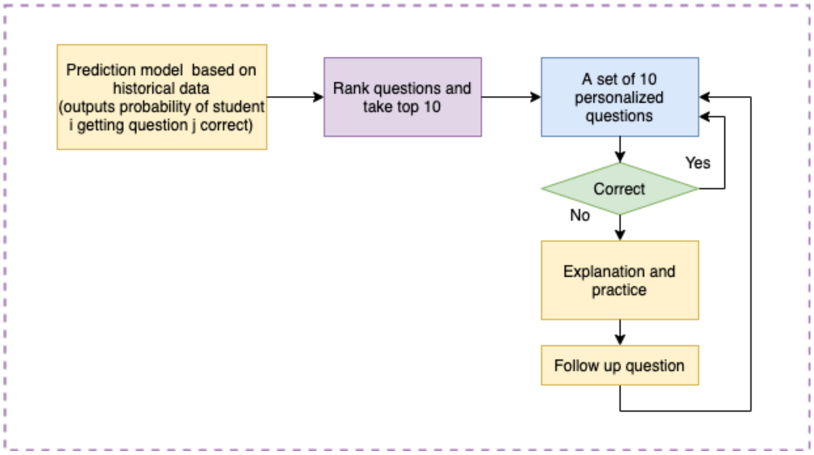
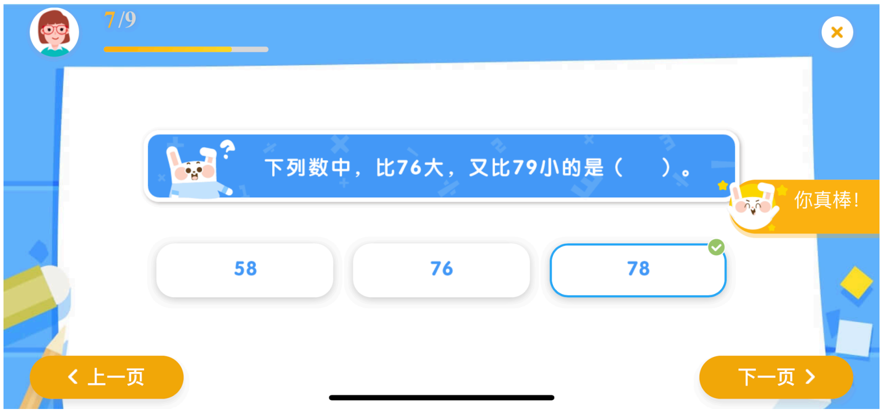
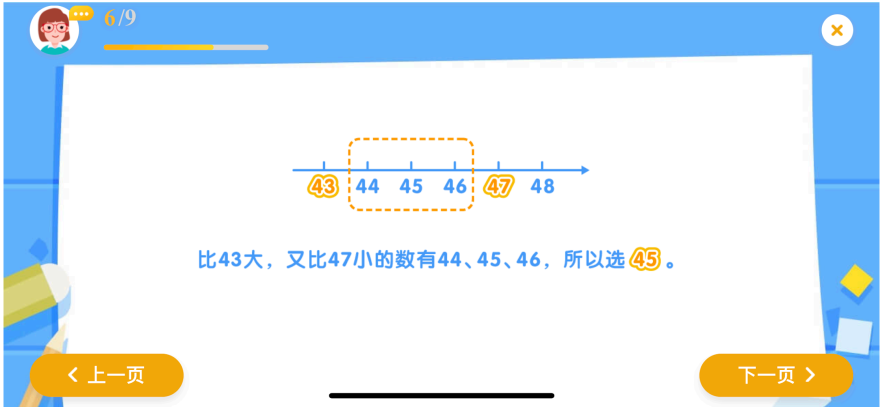

Prediction model based on historical data (outputs probability of student i getting question j correct)
Rank questions and take top 10
A set of 10 personalized questions
Correct
Yes
No
Explanation and practice
Follow up question

Figure 3.16: *Smart Homework* is powered by a recommeder system and immediate feedback



下列数中，比76大，又比79小的是（　）。

Figure 3.17: Question in an LPE module



比43大，又比47小的数有44、45、46，所以选45。

Figure 3.18: Explanatory text/audio in an LPE module

Our features load pre-existing parameters into the model in the form of prior distributions. The prior distributions encapsulate all the information the model learned from previous data. Thus, instead of re-fitting the model every time new data comes in, the model needs only to use incremental data to update itself using the prior distributions as its starting point. This helps reduce the computational costs of keeping models up-to-date. In practice, our model updates itself with incremental data once a week.

Secondly, we added a feature to allow parameters to be fixed. The fitting of the model will only alter trainable parameters but not fixed parameters. This allows question parameters, typically estimated over a large amount of historical data, to remain unchanged in the model fitting process. This lessens the burden of optimization and can further reduce computational costs when a large number of questions are involved.

3.9.2 Future experimentation

In a future study, we plan to experimentally test whether *Smart Homework* improves student outcomes. Namely, we will assign three versions of *Smart Homework* as the treatment condition and compare its effects on student engagement and achievement against a control group that received a curated (but non-personalized) homework. The three versions of *Smart Homework* will be 1) the full version with both personalization and feedback, 2) a partial version with only personalization, and 3) a partial version with only feedback.

Our partner company is currently (as of May 2021) conducting a pilot of *Smart Homework* with 30K student participants. The pilot stage aims to reveal bugs and inefficiencies in the product's engineering. The pilot stage also aims to establish an understanding of the baseline student behaviors including open rate, completion rate, and utilization of feedback. The main study will proceed after the issues revealed in the pilot stage are addressed.

3.10 Conclusion

We have shown that latent factorization and hierarchical modeling powered by Bayesian variational inference can make important gains in modeling student-question interactions using large datasets from online education. This set of modern approaches perform well in parameter retrieval and predictive accuracy. They are also numerically stable and can be applied to industrial-scale datasets with computational ease. Our models compare favorably in both computational performance and predictive accuracy against similar models from recent literature. Finally, they also allow for desirable features such as batch updates and parameter freeze to be implemented with ease.

Smart Homework is only one of the many applications that our models can power. Our models can be used to identify learning gaps. In particular, our factorization approach can identify content domains where a student is weaker than average. Early warning systems and remedial content recommendations can be generated based on such information.

Our models can also improve the informational landscape of parents and teachers. One example can be building learning progress dashboards. For this application, the predictions from our models will be used to derive statistics about student progress. An example of this is using trained models to make predictions about the average rate at which the student will get a particular set of pre-selected domain-representative questions correct. This statistic can be used as an indicator for proficiency.

Finally, applications can make use of the latent parameters the models learned from data. One such application is the automatic labeling of questions based on the latent parameters produced by our models. Another application is discovering the preferences of teachers in terms of which types of question a particular teacher likes to assign.

3.11 Appendix

3.11.1 Bayesian Variational Inference

The main method of inference for parameters in our proposed models is Bayesian variational inference (see (Blei et al., 2017) for a review). Bayesian variational inference is increasingly popular in estimation tasks involving large amounts of data. It has superior runtime performance compared to traditional Bayesian estimation techniques such as Markov Chain Monte Carlo. Variational inference has been shown to work well in mean-fields approximation tasks, but existing methods are less well suited to recover correlations between parameters.

Variational inference proposes a parameterized class of posterior distributions and reduces the Bayesian inference task to one of optimizing a divergence value between the proposed posterior and the true posterior. For this inference task, I use the Kullback-Leibler divergence defined by

$$\begin{aligned}\lambda^* &= \arg\min_{\lambda} KL(q(\boldsymbol{z};\lambda)||p(\boldsymbol{z}|\boldsymbol{x})) \\ &= \arg\min_{\lambda} E_{q(\boldsymbol{z};\lambda)}[\log(q(\boldsymbol{z};\lambda)) - \log(p(\boldsymbol{z}|\boldsymbol{x}))]\end{aligned}$$

where $\boldsymbol{z}$ is the parameter vector of interest, $\boldsymbol{x}$ is the data, and λ is the parameter vector characterizing the proposed posterior distribution. However, directly optimizing this expression is not possible since it involves the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$. However, since we can reformulate the divergence as

$$KL(q(\boldsymbol{z};\lambda)||p(\boldsymbol{z}|\boldsymbol{x})) = \log(p(\boldsymbol{x})) - \text{ELBO}(\lambda)$$

where ELBO(λ) is the evidence lower bound defined by

$$\text{ELBO}(\lambda) = E_{q(\boldsymbol{z};\lambda)}[\log(p(\boldsymbol{x},\boldsymbol{z})) - \log(q(\boldsymbol{z};\lambda))]$$

we can derive gradients to perform optimization on the ELBO.

Bibliography

Athey, S., Blei, D., Donnelly, R., Ruiz, F., and Schmidt, T. (2018). Estimating heterogeneous consumer preferences for restaurants and travel time using mobile location data. *arXiv preprint arXiv:1801.07826*.

Bailey, M. A. (2007). Comparable preference estimates across time and institutions for the court, congress, and presidency. *American Journal of Political Science*, 51(3):433–448.

Banerjee, A. V., Cole, S., Duflo, E., and Linden, L. (2007). Remedying education: Evidence from two randomized experiments in india. *The Quarterly Journal of Economics*, 122(3):1235–1264.

Bergman, P. (2015). Parent-child information frictions and human capital investment: Evidence from a field experiment.

Bergman, P. (2019). How behavioral science can empower parents to improve children's educational outcomes. *Behavioral Science & Policy*, 5(1):52–67.

Bergman, P., Lasky-Fink, J., and Rogers, T. (2019). Simplification and defaults affect adoption and impact of technology, but decision makers do not realize it. *Organizational Behavior and Human Decision Processes*.

Bettinger, E. P., Long, B. T., Oreopoulos, P., and Sanbonmatsu, L. (2012). The Role of Application Assistance and Information in College Decisions: Results from the H
amp;R Block Fafsa Experiment*. *The Quarterly Journal of Economics*, 127(3):1205–1242.

Bhatia, R. (2017). A look at how Byju's personalization engine is driving one-on-one learning experience.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

Bollen, D., Knijnenburg, B. P., Willemsen, M. C., and Graus, M. (2010). Understanding choice overload in recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 63–70.

Breman, A. (2011). Give more tomorrow: Two field experiments on altruism and intertemporal choice. *Journal of Public Economics*, 95(11-12):1349–1357.

Bryan, G., Karlan, D., and Nelson, S. (2010). Commitment devices. *Annu. Rev. Econ.*, 2(1):671–698.

Candes, E. J. and Plan, Y. (2010). Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936.

Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278.

Cortes, K. E., Fricke, H., Loeb, S., Song, D. S., and York, B. N. (2018). Too little or too much? actionable advice in an early-childhood text messaging experiment. *Education Finance and Policy*, pages 1–44.

Cunha, N., Lichand, G., Madeira, R., and Bettinger, E. (2017). What is it about communicating with parents.

Donnelly, R., Kanodia, A., and Morozov, I. (2020). A unified framework for personalizing product rankings. *Available at SSRN 3649342*.

Donnelly, R., Ruiz, F. R., Blei, D., and Athey, S. (2019). Counterfactual inference for consumer choice across many product categories. *arXiv preprint arXiv:1906.02635*.

Fricke, H., Kalogrides, D., and Loeb, S. (2018). It' s too annoying: Who drops out of educational text messaging programs and why. *Economics letters*, 173:39–43.

Gallego, F., Malamud, O., and Pop-Eleches, C. (2017). Parental monitoring and children's internet use: The role of information, control, and cues. Technical report, National Bureau of Economic Research.

Gopalan, P., Hofman, J. M., and Blei, D. M. (2015). Scalable recommendation with hierarchical poisson factorization.

Hernández-Blanco, A., Herrera-Flores, B., Tomás, D., and Navarro-Colorado, B. (2019). A systematic review of deep learning approaches to educational data mining. *Complexity*, 2019.

Imai, K., Lo, J., Olmsted, J., et al. (2016). Fast estimation of ideal points with massive data. *American Political Science Review*, 110(4):631–656.

Karr-Wisniewski, P. and Lu, Y. (2010). When more is too much: Operationalizing technology overload and exploring its impact on knowledge worker productivity. *Computers in Human Behavior*, 26(5):1061–1072.

Khajah, M., Lindsey, R. V., and Mozer, M. C. (2016). How deep is knowledge tracing? *arXiv preprint arXiv:1604.02416*.

Khan Academy (2020a). Khan Academy 2019 Annual Report.

Khan Academy (2020b). Using Khan Academy for personalized practice and mastery.

Khan Academy (2020c). Using Khan Academy for self-paced practice.

Kim, J. H., Kim, M., Kwak, D. W., and Lee, S. (2019). Assisting teachers with artificial intelligence: Investigating the role of teachers using a randomized field experiment. *Available at SSRN 3399851*.

Kolen, M. J. and Brennan, R. L. (2004). Test equating, scaling, and linking.

Konstan, J. A. and Riedl, J. (2012). Recommender systems: from algorithms to user experience. *User modeling and user-adapted interaction*, 22(1-2):101–123.

Lai, F., Zhang, L., Qu, Q., Hu, X., Shi, Y., Boswell, M., and Rozelle, S. (2012). Does computer-assisted learning improve learning outcomes? evidence from a randomized experiment in public schools in rural minority areas in qinghai, china. *Rural Education Action Project Working paper*, 237.

Mayer, S. E., Kalil, A., Oreopoulos, P., and Gallegos, S. (2015). Using behavioral insights to increase parental engagement: The parents and children together (pact) intervention. Technical report, National Bureau of Economic Research.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322.

Mo, D., Zhang, L., Luo, R., Qu, Q., Huang, W., Wang, J., Qiao, Y., Boswell, M., and Rozelle, S. (2014). Integrating computer-assisted learning into a regular curriculum: Evidence from a randomised experiment in rural schools in shaanxi. *Journal of development effectiveness*, 6(3):300–323.

Muralidharan, K., Singh, A., and Ganimian, A. J. (2019). Disrupting education? experimental evidence on technology-aided instruction in india. *American Economic Review*, 109(4):1426–60.

Pardos, Z. A. (2017). Big data in education and the models that love them. *Current opinion in behavioral sciences*, 18:107–113.

Pardos, Z. A. and Heffernan, N. T. (2011). Kt-idem: Introducing item difficulty to the knowledge tracing model. In *International conference on user modeling, adaptation, and personalization*, pages 243–254. Springer.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

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

Poole, K. T., Lewis, J. B., Lo, J., and Carroll, R. (2008). Scaling roll call votes with w-nominate in r. *Available at SSRN 1276082*.

Pu, P. and Chen, L. (2006). Trust building with explanation interfaces. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 93–100.

Reardon, S. F., Kalogrides, D., and Ho, A. D. (2019). Validation methods for aggregate-level test scale linking: A case study mapping school district test score distributions to a common scale. *Journal of Educational and Behavioral Statistics*, page 1076998619874089.

Rizopoulos, D. (2006). ltm: An r package for latent variable modeling and item response theory analyses. *Journal of statistical software*, 17(5):1–25.

Ruiz, F. J., Athey, S., and Blei, D. M. (2017). Shopper: A probabilistic model of consumer choice with substitutes and complements. *arXiv preprint arXiv:1711.03560*.

Shor, B. and McCarty, N. (2011). The ideological mapping of american legislatures. *American Political Science Review*, pages 530–551.

Singh, M. (2020). Indian education startup Byju's is fundraising at a $10B valuation.

Sunny Education Inc. (2018). 17ZUOYE Raises US$250 Million to Consolidate K-12 Edtech Market Leader Position in China.

Thaler, R. H. and Benartzi, S. (2004). Save more tomorrow™: Using behavioral economics to increase employee saving. *Journal of political Economy*, 112(S1):S164–S187.

Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., and Blei, D. M. (2016). Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*.

Vygotsky, L. S. (1980). *Mind in society: The development of higher psychological processes*. Harvard university press.

Wilson, K. H., Karklin, Y., Han, B., and Ekanadham, C. (2016a). Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. *arXiv preprint arXiv:1604.02336*.

Wilson, K. H., Xiong, X., Khajah, M., Lindsey, R. V., Zhao, S., Karklin, Y., Van Inwegen, E. G., Han, B., Ekanadham, C., Beck, J. E., et al. (2016b). Estimating student proficiency: Deep learning is not the panacea. In *In Neural Information Processing Systems, Workshop on Machine Learning for Education*, page 3.

Wu, M., Davis, R. L., Domingue, B. W., Piech, C., and Goodman, N. (2020). Variational item response theory: Fast, accurate, and expressive. *arXiv preprint arXiv:2002.00276*.

Yudelson, M. V., Koedinger, K. R., and Gordon, G. J. (2013). Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, pages 171–180. Springer.